

File No. S360-30  
Order No. GY28-6606-3

## **Program Logic**

### **IBM System/360 Operating System Catalog Management Program Logic Manual**

**Program Number 360S-DM-508**

This book describes the internal logic of the catalog-management routines. This information is directed to maintenance personnel and system programmers who require an in-depth knowledge of the program's design, organization, and data areas. It is not required for effective use of catalog management facilities. Flowcharts are included.

#### **Fourth Edition (December 1971)**

This edition corresponds to release 21 of the IBM System/360 Operating System. It is a major revision of, and makes obsolete:

- *IBM System/360 Operating System; Catalog Management; Program Logic Manual*, Order Number GY28-6606-2, with Technical Newsletters GN26-8006 and GY35-0002.
- *IBM System/360 Operating System; Time Sharing Option; Catalog Management; Program Logic Manual*, Order Number GY28-6745-0.

This publication has been entirely rewritten to reflect the total revision of the programs that it describes. With release 21, the catalog-management routines:

- Have been repackaged for better performance and maintainability.
- Can automatically create or delete catalog indexes.
- Use a different technique for ENQ/DEQ processing to improve catalog availability.
- Include the relative track address (TTR) of the DSCB for data sets that reside on one volume.
- Check for unsuccessful DOS-to-OS VTOC conversion.

The information contained in this publication is subject to significant change. Any such change will be reported in subsequent editions or technical newsletters. Before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/360 and System/370 SRL Newsletter*, Order Number GN20-0360, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office that serves you.

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the publication to: IBM Corporation, Programming Publications, Department 27D, Post Office Box 1900, Boulder, Colorado 80302. Comments become the property of IBM.

## Preface

This book describes the internal logic of the catalog-management routines and provides diagnostic information. This information is directed to maintenance personnel and development programmers who require an in-depth knowledge of the program's design, organization, and data areas. It is not required for effective use of catalog management facilities.

You should be familiar with general programming techniques, OS concepts and use, and System/360 before reading this book. If you are unfamiliar with the general concepts of catalog management, read:

- *OS Data Management Services Guide*, Order Number GC26-3746, which discusses the general concepts of data sets and catalog management.
- *OS Utilities*, Order Number GC28-6586, which contains information about IEHPROGM and generation data groups.
- *OS Data Management for System Programmers*, Order Number GC28-6550, which describes how catalog-management routines are invoked.

Other books that may be helpful to you are:

- *Guide to Reading OS System Dumps*, Order Number GC28-6670, which describes how to analyze a main storage dump from the operating system.
- *OS Service Aids*, Order Number GC28-6719, which describes several service aids and programs available under the operating system, notably for this book, IMASPZAP.
- *OS DADSM Logic*, Order Number GY28-6607, which provides details about module IGG0553A and rotational positioning support (RPS).
- *Introduction to OS Control Program Logic*, Order Number GY28-6605, which provides an overview of operating system logic.
- *OS SAM Logic*, Order Number GY28-6604, which provides details about the BLDL routine.
- *OS Control Blocks*, Order Number GC28-6628, which shows the content of most of the operating system control blocks and tables.

This book is divided into six chapters and a glossary:

1. "Introduction" describes what the catalog is, how the catalog is structured, and how the catalog-management routines are invoked. The chapter defines terms that are used throughout the book.
2. "Method of Operation" provides the design overview. This chapter describes the philosophy behind the programs. Emphasis is on the flow of data and the concepts of catalog management, rather than the organization of the modules.
3. "Program Organization" describes each module of catalog management and identifies the specific function that each performs to achieve the catalog management objectives. This chapter shows the logical flow from module to module and contains the flowcharts of the modules.
4. "Microfiche Directory" contains a table that relates the information in this book to the listings on microfiche.
5. "Data Areas" describes the record formats and work areas that are used by catalog management.
6. "Diagnostic Aids" can help you when you're diagnosing a problem. This chapter shows how to dump and analyze the catalog, and how to determine what modules are used for a particular case.
7. A glossary of terms and acronyms used in this publication is found just before the index.



# Contents

<b>Summary of Amendments</b> . . . . .	7
<b>Introduction</b> . . . . .	9
SYSCTLG Data Set . . . . .	9
Names and Indexes . . . . .	9
Control Volumes . . . . .	10
Aliases . . . . .	11
Generation Data Groups . . . . .	11
Entries of a Catalog . . . . .	11
Using the Catalog-Management Routines . . . . .	12
<b>Method of Operation</b> . . . . .	14
Operation I, Read . . . . .	14
Operation II, Set-up . . . . .	17
Operation III, Write . . . . .	17
<b>Program Organization</b> . . . . .	21
Overall Organization . . . . .	21
Macros and Services Used . . . . .	23
Character Dependency . . . . .	24
Resource Enqueuing . . . . .	24
Register Usage . . . . .	24
Module Descriptions . . . . .	24
IGC0002F: Initialization . . . . .	26
IGG0CLC1: Relative GDG and Alias . . . . .	28
IGG0CLC2: Locate . . . . .	30
IGG0CLC3: Update Initialization and Entry Building . . . . .	32
IGG0CLC4: Entry Building . . . . .	34
IGG0CLC5: First Load of Update . . . . .	36
IGG0CLC6: Second Load of Update . . . . .	38
IGG0CLC7: Third Load of Update and Error Handling . . . . .	40
IGC0002H: SYSCTLG Open/Extend . . . . .	42
IGG0CLF2: SYSCTLG and BPAM Directory Formatter . . . . .	44
<b>Microfiche Directory</b> . . . . .	46
<b>Data Areas</b> . . . . .	47
SYSCTLG Entry Formats . . . . .	47
Alias Entry (AE) . . . . .	48
Control Volume Pointer Entry (CVPE) . . . . .	48
Old CVOL Pointer Entry . . . . .	48
New CVOL Pointer Entry . . . . .	48
Data Set Pointer Entry (DSPE) . . . . .	49
Generation Index Pointer Entry (GIPE) . . . . .	49
Index Control Entry (ICE) . . . . .	50
Index Link Entry (ILE) . . . . .	50
Index Pointer Entry (IPE) . . . . .	50
Volume Control Block (VCB) . . . . .	51
Volume Control Block Pointer Entry (VCBPE) . . . . .	51
Volume Index Control Entry (VICE) . . . . .	52
Environment Record (EREC DSECT) . . . . .	52
CAMLSTDT . . . . .	53
RPSD DSECT . . . . .	54
WORKAREA DSECT . . . . .	54
<b>Diagnostic Aids</b> . . . . .	60
Reading Dumps . . . . .	60
Main Storage Dump . . . . .	60
SYSCTLG Dump . . . . .	60
Example of a SYSCTLG Dump . . . . .	61
Environment Record . . . . .	62
Module Selection Charts . . . . .	62
<b>Glossary</b> . . . . .	64
<b>Index</b> . . . . .	66

<b>Figures</b>	Figure 1. Example of a Catalog with One Simple Name . . . . .	10
	Figure 2. Example of Catalog with a Qualified Name . . . . .	10
	Figure 3. Example of Catalog with Alias and Connected CVOL . . . . .	11
	Figure 4. Example of Catalog with a Generation Data Group . . . . .	12
	Figure 5. Kinds of Entries in the Catalog . . . . .	12
	Figure 6. Overall Program Organization . . . . .	22
	Figure 7. Resource Enqueueing . . . . .	25
	Figure 8. DCB/DEB Constructed by IGC0002H . . . . .	42
	Figure 9. Data Area Hierarchy . . . . .	54
	Figure 10. Example of a SYSCTLG Dump . . . . .	61
	Figure 11. Correlating Functions to Modules of Catalog Management . . . . .	63
<b>Method of Operation Diagrams</b>	Diagram MO1. Catalog Management Overview . . . . .	15
	Diagram MO2. Operation I—Read . . . . .	16
	Diagram MO3. Operation II—Setup . . . . .	18
	Diagram MO4. Operation III—Write . . . . .	19
<b>Flowcharts</b>	Flowchart FC1. IGC0002F: Initialization . . . . .	26
	Flowchart FC2. IGG0CLC1: Relative GDG and Alias . . . . .	28
	Flowchart FC3. IGG0CLC2: Locate . . . . .	30
	Flowchart FC4. IGG0CLC3: Update Initialization and Entry Building . . . . .	32
	Flowchart FC5. IGG0CLC4: Entry Building . . . . .	34
	Flowchart FC6. IGG0CLC5: First Load of Update . . . . .	36
	Flowchart FC7. IGG0CLC6: Second Load of Update . . . . .	38
	Flowchart FC8. IGG0CLC7: Third Load of Update and Error Handling . . . . .	40
	Flowchart FC9. IGC0002H: SYSCTLG Open/Extend . . . . .	42
	Flowchart FC10. IGG0CLF2: SYSCTLG and BPAM Directory Formatter . . . . .	44

## Summary of Amendments for OS Release 21

<b>Total Replacement of Programs and Publications</b>	The catalog management routines have been completely rewritten. The organization of the new routines into modules is entirely different from the previous modules. Similarly, the publication has also been completely rewritten and its organization has no relationship to previous editions.
<b>Create a Delete Indexes</b>	Two new functions have been added to catalog management: CATBX and UCATDX. CATBX can be used to simultaneously catalog a data set and build any index levels that are required. UCATDX can be used to simultaneously uncatalog a data set and delete index levels.
<b>Data Set Pointer Entry</b>	The relative track address (TTR) of the DSCB for a data set that resides on one volume is now included in the catalog's Data Set Pointer Entry for that data set.
<b>DOS-to-OS Conversion</b>	Catalog management routines now check for an unsuccessfully converted VTOC (DOS to OS).





## Introduction

Catalog management is the part of the IBM System/360 Operating System that enables a user to locate a data set by specifying only a data set name. Catalog management stores data set names with volume and device information in a catalog, itself a data set named SYSCTLG. In use, the name in the user's request is correlated with the appropriate volume identification. The catalog-management routines organize the catalog by inserting, removing, and locating entries in the catalog, and by formatting new catalogs.

Catalog management is a type 4 SVC. The catalog-management routines execute in the 1024-byte transient areas of the operating system nucleus. System and machine requirements, therefore, are those of the operating system.

This chapter discusses the catalog, illustrates several examples of a catalog, and discusses how the catalog-management routines are invoked.

### SYSCTLG Data Set

The catalog is always named SYSCTLG; thus, there can be only one OS Catalog on any one volume.

SYSCTLG data sets are formatted like partitioned data set (PDS) directories. Each physical block consists of an 8-byte key and a 256-byte data record. The format of the logical records contained in the physical block is shown in the chapter "Data Areas," page 47.

### Names and Indexes

Every cataloged data set has a name; this name can be a *simple name* or a *qualified name*.

A simple name is a collection of up to eight EBCDIC characters. A qualified name is a string of names, separated by periods. The last name in the string is a simple name; the preceding names are *qualifiers*. Each qualifier can be up to eight characters long; the maximum total length of a qualified name is 44 characters.

Each qualifier of a data set name corresponds to one *index* in the catalog. The structure of these indexes forms the catalog. The simple name of a data set is found in the lowest level of structure corresponding to the fully qualified name. The highest level of the catalog is called the *volume index*; the volume index is automatically built the first time a new catalog is used by the catalog-management routines.

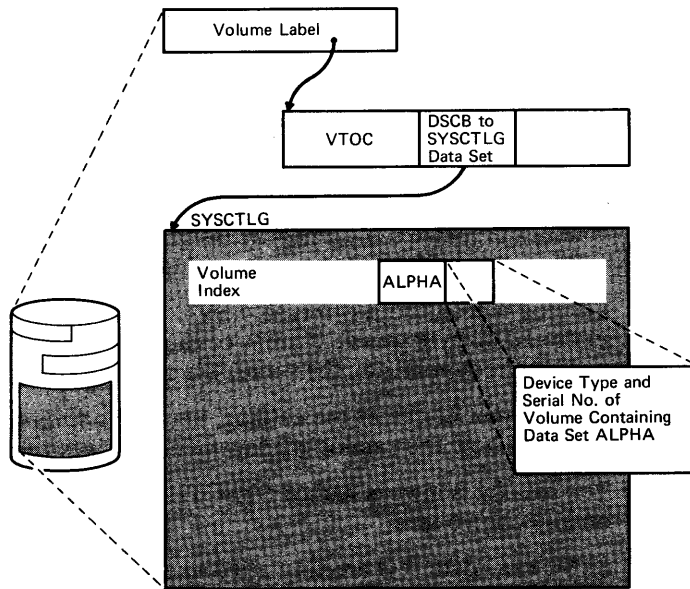
Names in an index level are arranged in EBCDIC collating sequence.

Each physical block of SYSCTLG can contain only one index level. An index level, however, can span more than one block. Multiple blocks of one index are chained together with the last logical record in each block. Contiguous blocks make use of the key to speed searching.

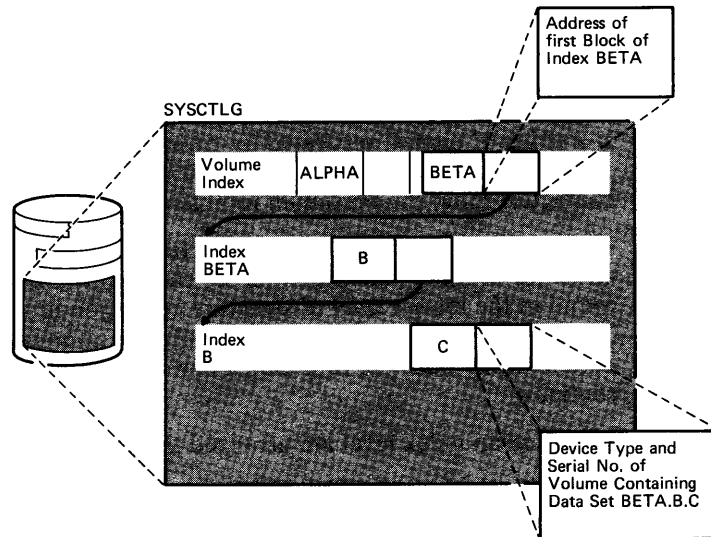
Figure 1 shows an example of a catalog containing only the volume index and one simple name. (We will build on this figure throughout this chapter.) As with any data set on a direct access volume, SYSCTLG is found by tracing through the Volume Label to the Volume Table of Contents (VTOC). The VTOC contains a Data Set Control Block (DSCB) for each data set on that volume (independent of whether that data set is cataloged). The Format 1 DSCB for SYSCTLG points to the catalog data set.

The catalog data set is shown as the shaded block in Figure 1. The first record of SYSCTLG contains the volume index. The entry in that volume index identifies the device type and serial number for the volume that contains the data set ALPHA.

Figure 2 shows a qualified name in the catalog. Cataloging the data set BETA.B.C adds another entry to the volume index. This entry points to the BETA index, which contains an entry for B. Following the pointer to the B index, we find an entry for C, which identifies the volume containing data set BETA.B.C.



**Figure 1. Example of a Catalog with One Simple Name**



**Figure 2. Example of Catalog with a Qualified Name**

## Control Volumes

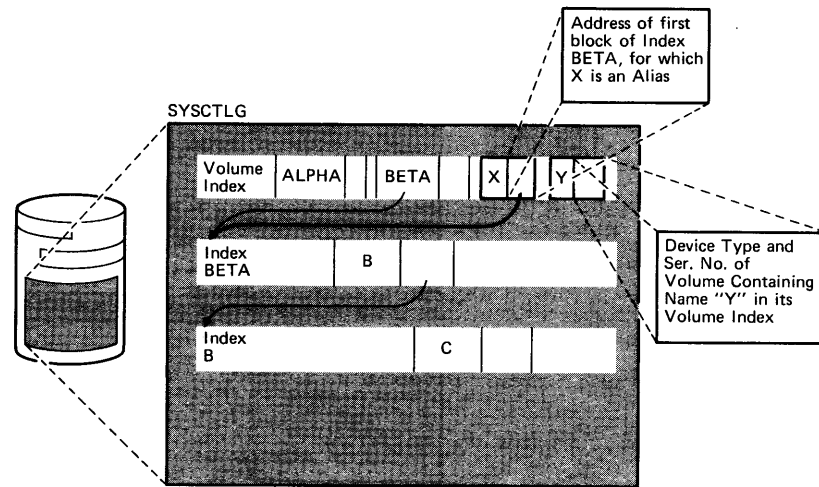
Any direct access volume except an IBM 2321 Data Cell may contain a catalog. A volume that contains a catalog is called a *control volume (CVOL)*. The system residence volume always contains a catalog.

A user specifies which CVOL contains the catalog (perhaps by allowing the default to the system residence volume). For a data set to be available through the catalog, its name must appear in either the catalog on the specified CVOL or in a CVOL that is *connected* to the specified CVOL. When a CVOL is connected, there is a pointer in the volume index of the catalog on the specified CVOL. This identifies the connected CVOL and names a high-level index on the connected CVOL.

A search for a data set may start on the system residence volume; a CVOL pointer on the system residence volume can direct the search to a connected CVOL. The search starts again on the connected CVOL, using the fully qualified name in the CVOL's volume index. The search may alternately be started on a CVOL other than the system residence volume, as directed by the user.

Figure 3 shows a CVOL pointer in the catalog. This CVOL pointer connects another CVOL to this catalog. In Figure 3, an entry for Y has been placed in the

volume index. This entry identifies the volume that contains the name Y in the volume index of its catalog.



**Figure 3. Example of Catalog with Alias and Connected CVOL**

**Aliases**

An *alias* is an alternative name for the highest-level index in a qualified data set name. There can be as many as 255 aliases to the first component of a qualified name.

Figure 3 also shows an alias entry. Here the name X is an alias for BETA, so an entry for X is placed in the volume index. The entry for BETA and the entry for X both point to index BETA. Note that only the high-level name can have an alias.

**Generation Data Groups**

A *generation data group* (GDG) is a collection of data sets that have a common external name, and that are related by the time sequence in which they were cataloged (that is, their generation). For example, a data set named LAB.PAYROLL(0) refers to the most recent data set of the group; LAB.PAYROLL(-1) refers to the second most recent data set, etc. The number in parentheses is called the *relative number* of a specific generation.

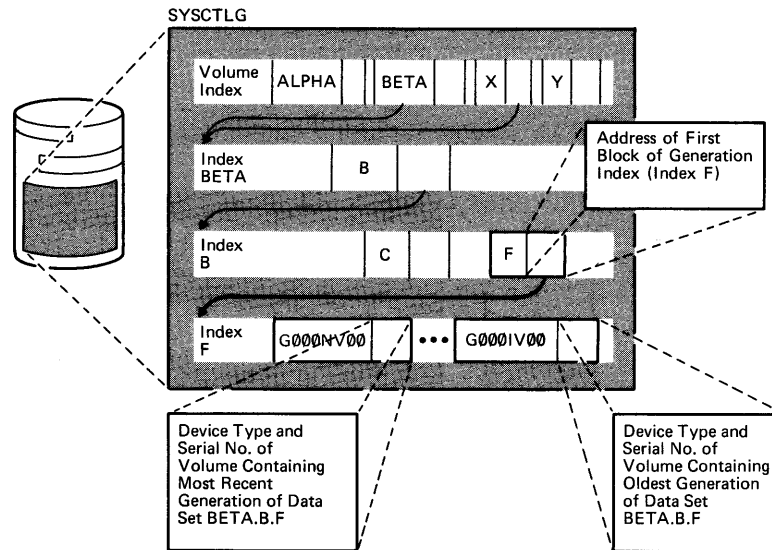
The relative number relates to an *absolute name* for that generation of the data set. The absolute name is the simple name of the data set. The absolute name has the general form G nnnn V mm, where nnnn is the generation number, and mm is the version number. The last qualifier in a relative GDG name is the name of the *generation index*. The generation number in the absolute name is complemented before the name is added to the generation index, so the most recent name (highest generation number) is the first entry in the index.

Figure 4 shows a generation data group in the catalog. In this example, the external data set name BETA.B.F applies to a generation data group, consisting of generations of the same data set. Here the lowest qualifying name, F, does not point to the data set, but rather points to the generation index. This generation index consists of pointers to generations of the data set group BETA.B.F.

**Entries of a Catalog**

The logical records of the SYSCTLG data set are called *entries*. These entries are linked together to form indexes and to connect indexes together. Figure 5 shows and names each type of entry.

Each index contains one control entry and any number of pointer entries. The control entry, which is the first entry on each index, identifies that index. Other entries in the index link blocks of one index together, point to lower indexes, and point to data sets.



**Figure 4. Example of Catalog with a Generation Data Group**

<i>Entry Name</i>	<i>Abbreviation</i>	<i>Description</i>
Alias entry	AE	Contains an alternate name for the high-level qualifier of a data set name.
Control volume pointer entry	CVPE	Connects another control volume (CVOL) to this CVOL; contains a name that appears in the volume index of the connected CVOL.
Data set pointer entry	DSPE	Contains the simple name of a data set and provides the location of this data set.
Generation index pointer entry	GIPE	Points to the lowest index for a generation data group.
Index control entry	ICE	First entry of all indexes other than volume index; describes the index level for one qualifier of a data set name.
Index link entry	ILE	Links this block to the next block in a chain of blocks for one index.
Index pointer entry	IPE	Points to a lower-level index of this name.
Volume control block	VCB	A block of the catalog that contains the volume list for a data set that resides on more than five volumes.
Volume control block pointer entry	VCBPE	Points to a volume control block.
Volume index control entry	VICE	The first entry in the volume index (the highest-level index in the catalog); describes the volume index and controls allocation of space in the catalog on one volume.

**Figure 5. Kinds of Entries in the Catalog**

## Using the Catalog-Management Routines

Catalog-management routines are invoked by the utility IEHPROGM, the job scheduler, the Time Sharing Option of the operating system (TSO), or by macro statements in an Assembler language program.

IEHPROGM creates and deletes indexes, aliases, and generation indexes, and it catalogs and uncatalogs data sets according to specifications supplied by the user of IEHPROGM.

The job scheduler calls the catalog management routines when it must locate a data set, given only its name, or when the DISP parameter of a DD statement implies catalog action.

Dynamic allocation with TSO locates old data sets and catalogs new data sets. TSO command processors also call the catalog management routines to manipulate the catalog.

All of these invocations of catalog management are produced by using three macro instructions: LOCATE, CATALOG, and INDEX. These macros generate a reference to a parameter list (produced by the CAMLST macro) and an SVC 26 instruction.

One of the operands of the CAMLST macro instruction specifies the function to be performed. These functions are categorized as *locate functions* or *non-locate functions*. The locate functions are those that require read-only operations to the catalog, that is, the catalog is not changed. The non-locate functions are those that do require changes to the catalog.

The locate functions are:

- NAME, which locates a data set by name.
- BLOCK, which locates a block of SYSCTLG.

The non-locate functions are:

- BLDX, which creates one index level.
- BLDG, which creates a generation index.
- BLDA, which assigns an alias to an existing high-level name.
- DLTX, which deletes one index level.
- DLTA, which deletes an alias.
- LNKX, which connects a CVOL.
- DRPX, which disconnects a CVOL.
- CAT, which adds a data set name to the catalog.
- CATBX, which adds a data set name to the catalog and creates any missing index levels.
- UNCAT, which removes a data set name from the catalog.
- UCATDX, which removes a data set name from the catalog and deletes index levels that are no longer needed (except the high-level index).
- RECAT, which recatalogs a data set.

For more information about the use of a catalog, see *OS Data Management for System Programmers*, Order Number GC28-6550 (the first chapter), or *OS Utilities*, Order Number GC28-6586 (the section on IEHPROGM).

For more information about generation data groups, see *OS Data Management Services Guide*, Order Number GC26-3746. Another discussion can be found in *OS Utilities*, in an appendix entitled “Generation Data Groups.”

## Method of Operation

This chapter explains the design of catalog management.

In its simplest form, catalog management consists of three operations: read, set-up, and write. Diagram MO1 illustrates this overview. Input to catalog management is shown in the block on the left, processing is shown in the center block, and output from catalog management is shown in the block to the right. The three operations are shown in the process block.

The input to catalog management consists of a parameter list built by the caller with the CAMLST macro instruction, supervisor control blocks, and the existing catalog. The output from catalog management depends on the requested function. Locate functions result in a volume list returned in the caller's area. Non-locate functions result in an updated catalog.

For a locate function, only the read operation is required. For a non-locate function, the read operation is used to locate the catalog block to which a change is to be made. The setup operation determines how the change is to be made, and the write operation updates the catalog accordingly.

The CATBX function is slightly more complex; it requires that set-up and write be executed several times. I'll discuss that function in more detail after describing each of the operations.

### Operation I, Read

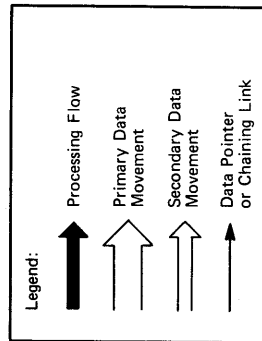
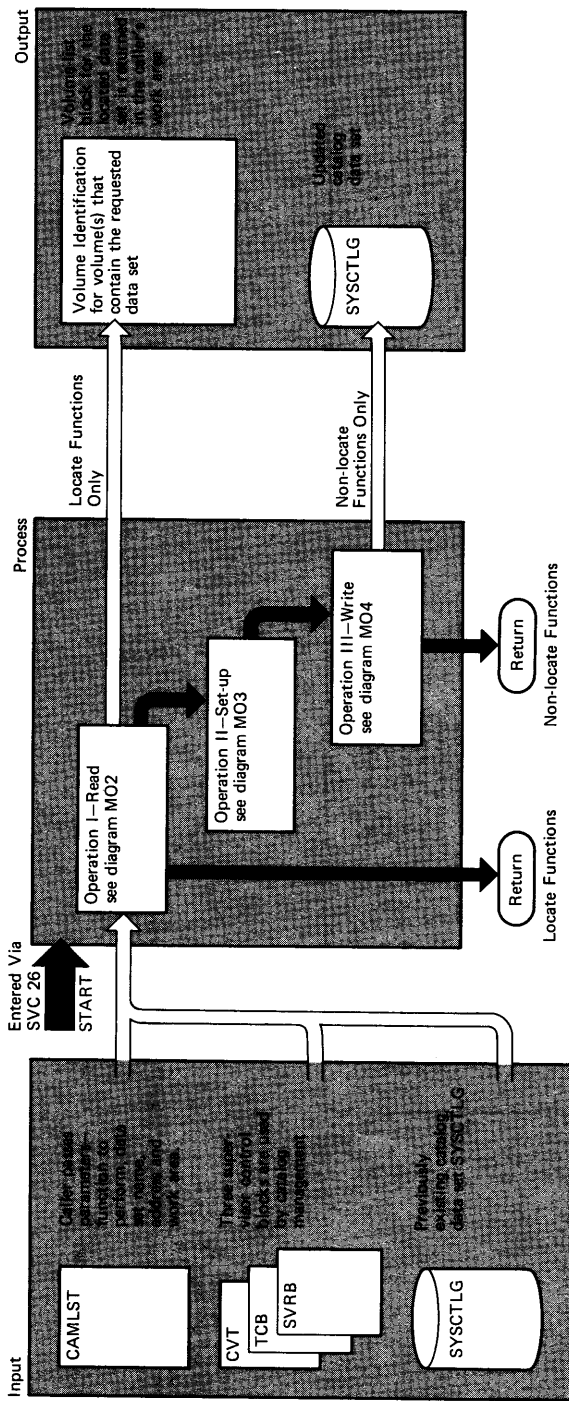
Diagram MO2 illustrates the concepts of Operation I, read. The input is that shown earlier in the overview. The caller's parameter list includes a function request, points to a data set name, and identifies the CVOL to use (possibly the system residence volume by default). If a function is a locate function, an area in the caller's module is made available for use as a return buffer. The usual SVC linkage for a type 3 SVC or type 4 SVC (catalog management is a type 4 SVC) provides pointers to several system control blocks. The VTOC of the CVOL is read to find the SYSCTLG data set and open it.

Operation I begins by initializing a work area named WORKAREA. The CVOL is opened by constructing a data control block (DCB) and a data extent block (DEB), which are shown as intermediate results in Diagram MO2. BLDL uses the DCB and DEB to read blocks of the catalog. The components of the name are isolated, and the high-level name is used as the search parameter of BLDL. BLDL begins by reading the block that contains the entry for the high-level name.

If the entry that is found is a CVOL pointer entry, the current CVOL is closed and the CVOL identified by the pointer is opened. The process repeats until the appropriate CVOL is open.

If the entry points to a lower level of the index structure, BLDL is called with the next component of the data set name as a search parameter. This process is repeated until the volume list is read (for locate functions), or the last valid level of the structure is found (for non-locate functions).

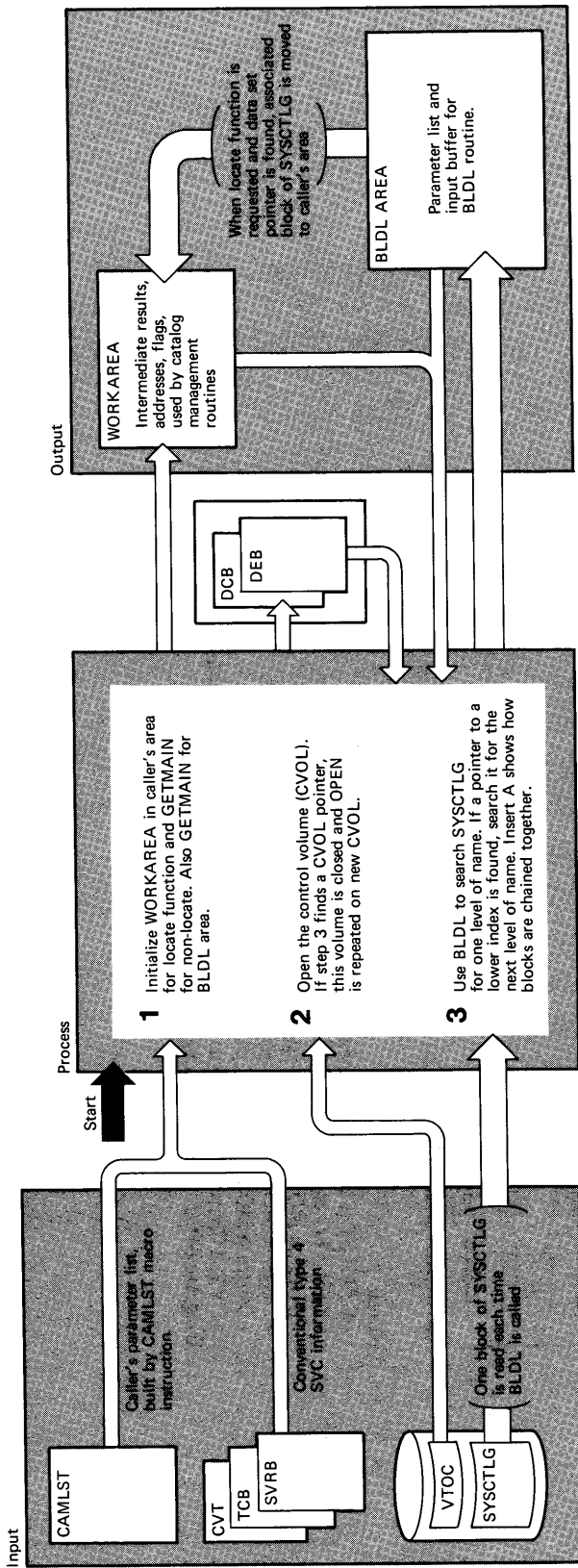
Insert A in diagram MO2 shows how the blocks of the catalog are chained together and how keys are used in this chaining process. The blocks are visualized in this insert as several contiguous blocks in the catalog, with block addresses ascending from left to right. The last entry of any catalog block is the index link entry (ILE), which is used to chain one block of an index to the next block in the same index. When two blocks are chained together but are not contiguous, as in the case of the first and fourth blocks in the insert for index A, the first block of the pair has a key equal to X'FFFFFFFFFFFFFFFF'. BLDL functions by first issuing a "search key high-or-equal" operation to the direct access device. When searching non-contiguous blocks, the high value of the key causes each block to be read and scanned for the requested name.



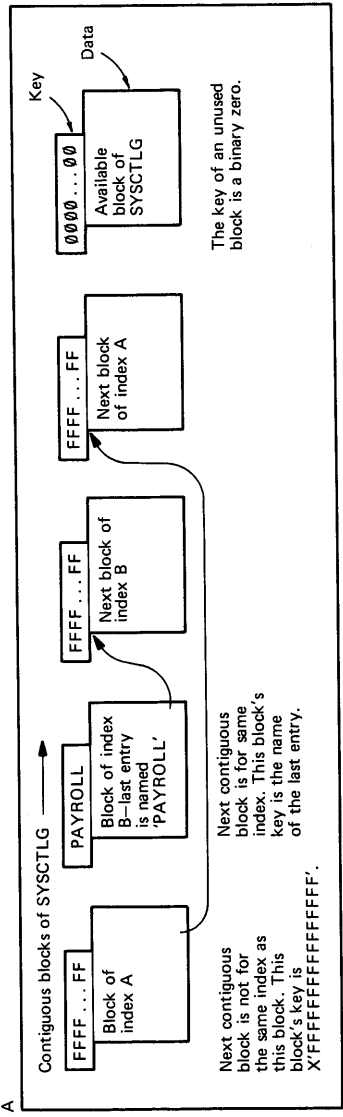
**Notes:**

- CAMLST is described on page 53.
- SYSCTLG entries are described beginning on page 47.
- Diagram MO2 is found on page 16.
- Diagram MO3 is found on page 18.
- Diagram MO4 is found on page 19.
- CVT, TCB, and SVRB are described in *OS Control Blocks*, Order Number GC28-6628.

**Diagram MO1. Catalog Management Overview**



See diagram MO1 to relate this process block to overall Method of Operation



**Notes:**

- CAMLST is described on page 53.
- WORKAREA is described on page 54.
- BLDLAREA is described on page 58.
- CVT, TCB, SVRB, VTOC, DCB, and DEB are described in *OS Control Blocks, Order Number GC28-6628*.

**Diagram MO2. Operation I—Read**



However, when two blocks are chained together and are contiguous, as is the case with index B with the second and third blocks of the insert, the first block of the pair has a key equal to the name in the next-to-last entry of that block. The “search key high-or-equal” operation stops on such a block only if the requested name could be in it; otherwise, the search continues with the next contiguous block.

Available blocks (that is, blocks that are unused) have a key equal to X‘0000000000000000’. Thus an available block of the catalog can be found by searching for a zero key.

## Operation II, Set-up

Diagram MO3 illustrates the setup operation. Input to this operation includes the name of an entry to be added or deleted from the catalog. This might be the simple name of a data set to be cataloged or removed from the catalog, or it might be an index name for a level to be created or deleted. Other information needed to build a new entry might be passed, depending on the request. The catalog block to which the change is to be made remains in a buffer from Operation I.

The setup process consists of either naming an entry to be deleted or constructing a new entry, then copying the catalog block one entry at a time from an input buffer to an output buffer. The new entry is merged into collating sequence with the old entries, or the named entry is bypassed, depending on the request.

Adding or deleting entries causes the following entries in the index to be displaced. For example, adding to a full block can result in an “overflow” of entries. These overflow entries must be added to the next block in the index chain. The effect of a change can thus *ripple* down the index chain. Similarly, a deletion can result in rippling in the other direction.

## Operation III, Write

Diagram MO4 illustrates the write operation. Input to the operation is an updated catalog block and a relative track address to which this block is to be rewritten into SYSCTLG. Rippling can cause several blocks to be rewritten. Finally, the block containing the index control entry (ICE) and the volume index control entry (VICE) are updated and rewritten to reflect the changes made to the catalog. Output is the updated catalog.

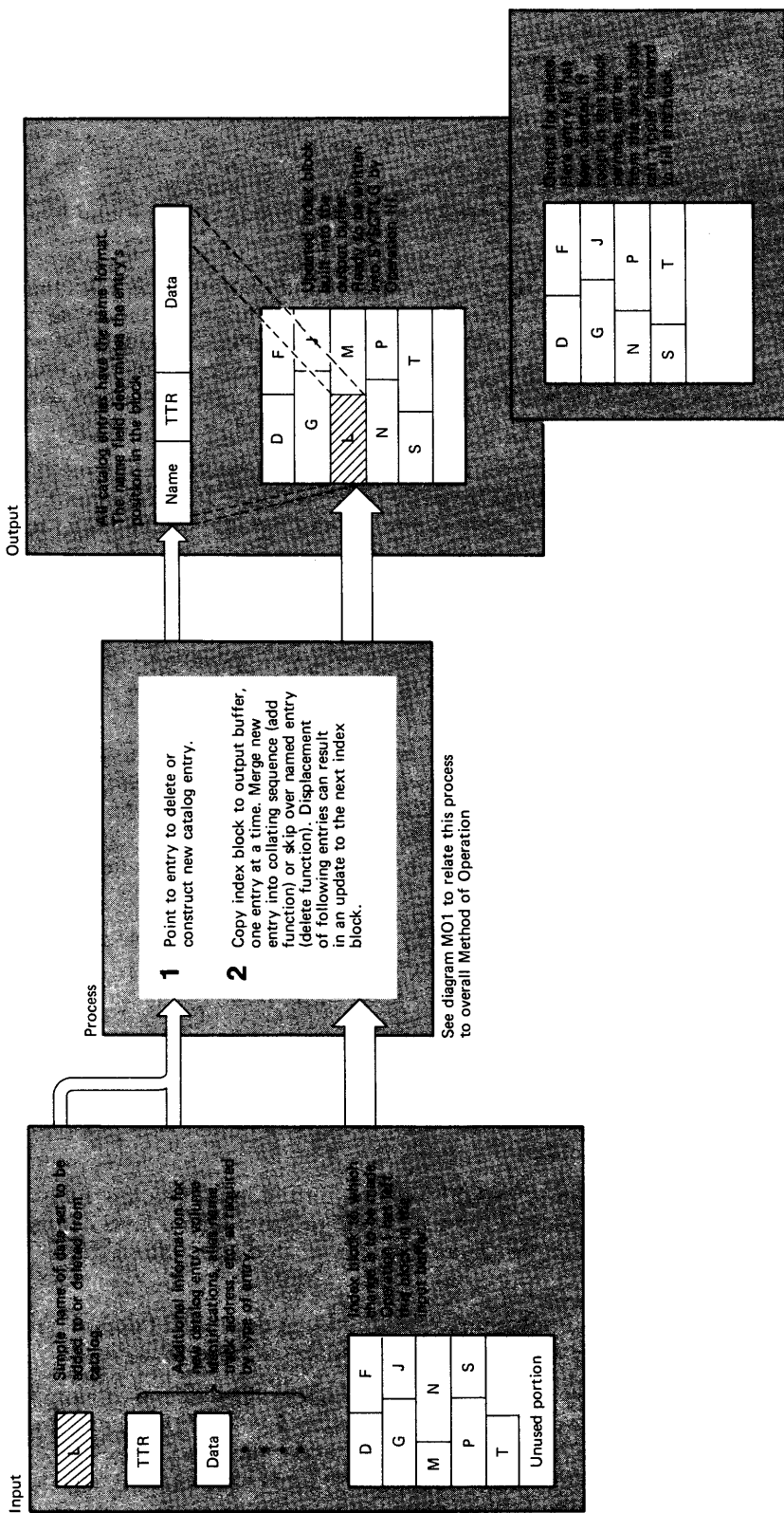
The insert to diagram MO4 shows how the CATBX function is performed. In this case, the index levels to which a data set name is to be added do not exist; these index levels must also be created. This is done from the bottom up.

Step 1 shows the new data set pointer entry (DSPE) ready to be added to the catalog. A new index block is constructed and this DSPE is inserted into it. The newly constructed index block is written to the first available space in SYSCTLG, and the relative track address of that block is saved for step 2. The catalog is searched for another available space.

In step 2, a new index pointer entry (IPE) is created for the index block written in step 1, using the relative track address from step 1. If the next higher level of the index structure is also missing, another index block is created and written to SYSCTLG.

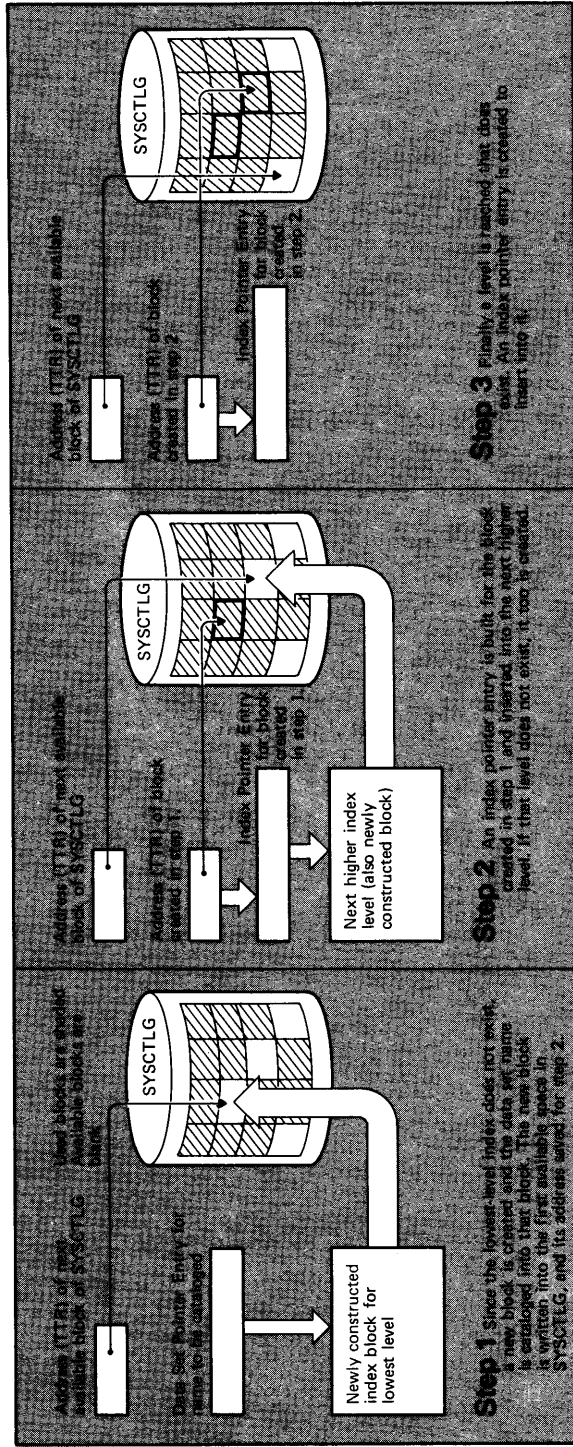
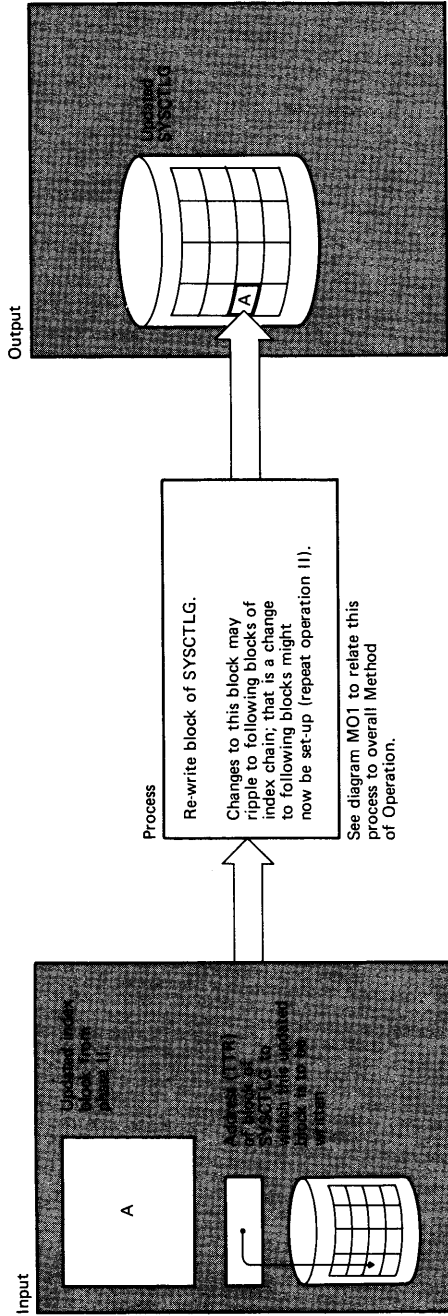
This process is repeated until a level is finally reached that does exist (step 3). An IPE is constructed for the highest new index and is added to the existing level.

For example, if the data set name A.B.C.DATASET were added to the catalog when none of the index levels A, B, or C exist, then step 1 would build index level C with a DSPE for A.B.C.DATASET in it. Step 2 would build index level B with an IPE for C in it, and then index level A with an IPE for B in it. Finally, step 3 would insert an IPE for A into the volume index.



- Notes:
- SYCTLG entries are described beginning on page 47.
  - BLDLAREA is described on page 58.
  - Diagram MO1 is found on page 15.

**Diagram MO3. Operation II—Setup**



- Notes:
- Diagram MO1 is found on page 15.
  - SYSCTLG entries are described beginning on page 47.

**Diagram MO4. Operation III—Write**



## Program Organization

This chapter discusses the implementation of the catalog management routines, showing how the actual modules fit together.

The final authority of any program is the assembly or compilation listing. These module discussions complement the assembly-listing comments, and assume that the listings are at hand. You should have them available for any in-depth analysis.

### Overall Organization

The three operations described in “Method of Operation” form a conceptual image of catalog-management. In reality, the broad operations of read, setup, and write merge together somewhat when packaged into modules. Still, the concept is useful, and it is continued in this chapter.

There are ten modules in catalog management. Eight of the modules form three phases, and two of the modules provide auxiliary services used by any phase.

Phase I equates to the read operation. It performs locate functions and the corresponding part of non-locate functions. Phase I consists of:

- IGC0002F, Initialization, which initializes work areas and opens the CVOL.
- IGG0CLC1, Relative GDG and Alias, which resolves aliases and relative GDG numbers.
- IGG0CLC2, Locate, which searches the lower levels of the index structure.

Phase II begins the setup operation. It checks the validity of the requests against the existing entries in SYSCTLG, and builds new entries to add or names entries to delete. Phase II consists of:

- IGG0CLC3, Update Initialization and Entry Building, which begins the update process by building new index blocks and routing the request as needed.
- IGG0CLC4, Entry Building, which builds data set pointer entries to add to the last valid level of the index.

Phase III completes the setup operation and performs the write operation. It merges entries into SYSCTLG blocks, deletes entries from blocks, and does most of the writing that is needed. Phase III consists of:

- IGG0CLC5, First Load of Update, which frees index blocks, frees volume control blocks (VCBs), and writes new VCBs.
- IGG0CLC6, Second Load of Update, which updates blocks, writes updated blocks to SYSCTLG, and ripples the change as needed to the last block of the updated chain.
- IGG0CLC7, Third Load of Update and Clean-up, which writes the last updated block, updates the control entries, and returns control to the caller. This module also handles error conditions.

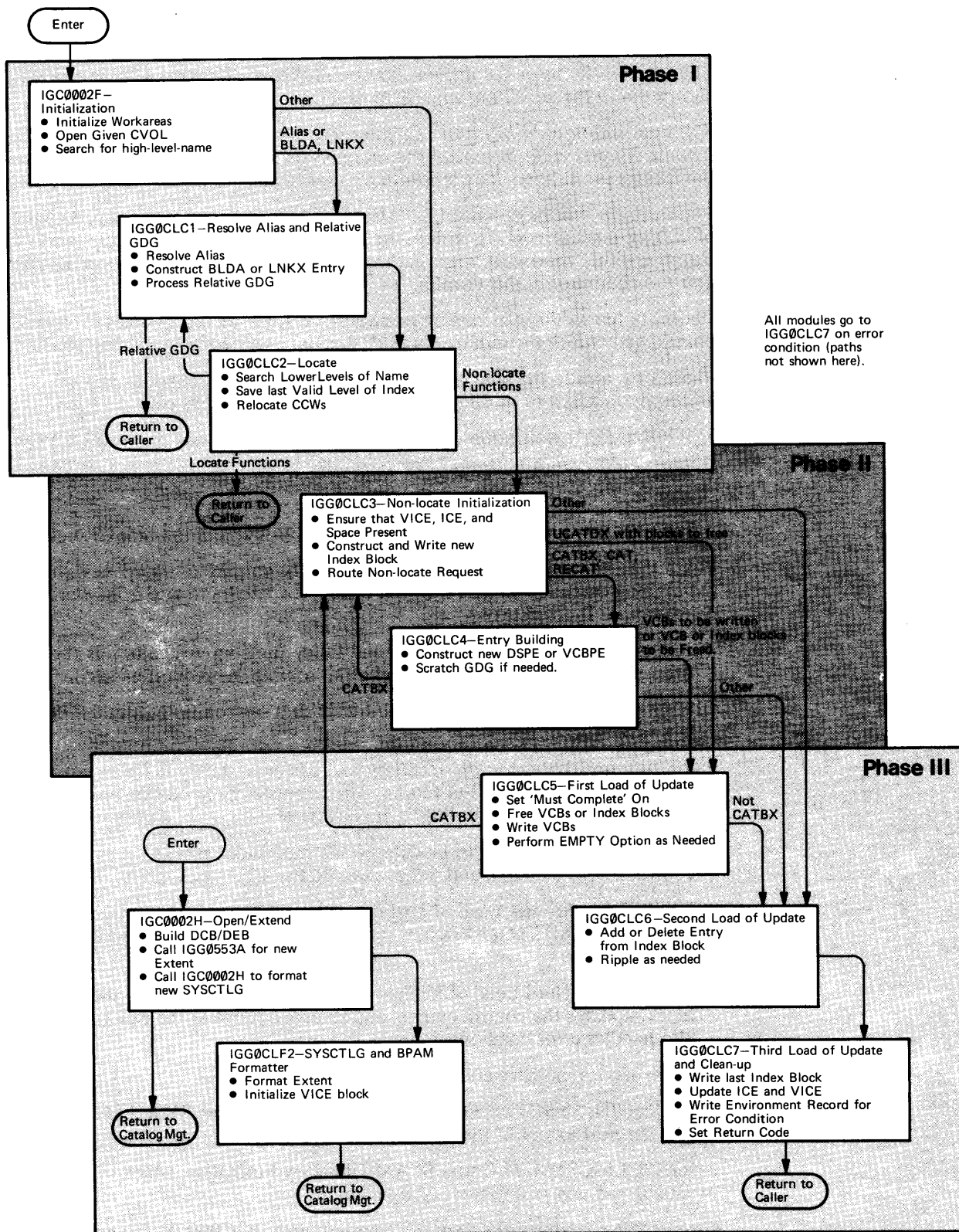
The two service modules are:

- IGC0002H, SYSCTLG Open/Extend, which opens the SYSCTLG data set or gets the next extent of that data set, when needed.
- IGG0CLF2, SYSCTLG and BPAM Directory Formatter, which formats a new catalog and any new extents of a catalog.

Each of these modules operates in a transient area. IGC0002F, the entry point for catalog management, is invoked by a caller with an SVC 26; control is passed to the other seven modules of the phases via an XCTL macro instruction.

IGC0002H, one of the service modules, is invoked via an SVC 28; it passes control to IGG0CLF2 via an XCTL macro instruction.

Figure 6 shows the organization of modules with relation to the phase structure and illustrates the overall flow of control from one module to another.



**Figure 6. Overall Program Organization**

An SVC 26 invokes the catalog-management routines; control is then passed to IGC0002F. IGC0002F is shown upper-left in Figure 6. The path that occurs through the remaining modules depends on both the particular function that is requested and the entries that are found in the catalog.

Each block in Figure 6 represents a module of the catalog-management routines and contains a brief description of the functions that it performs. Each path is identified by the function/condition that it represents.

For example, follow the path for a BLDX function (request to create an index level). Assume that part of the index structure already exists; that is, this request extends an existing index structure.

Specifically, this is what each module does to accomplish the BLDX request:

- Entry to catalog management is at IGC0002F, which initializes work areas, opens the CVOL, and locates the high-level name of the index structure. The arrow labeled “other” is the exit path for this example.
- IGG0CLC2 locates the remaining levels of the existing index structure, to find the last valid level. The new index is to be added to that level.
- BLDX is a non-locate function, so control passes to IGG0CLC3. This module reads the control entries (VICE and ICE), constructs an index pointer entry (IPE), and routes the request via the arrow labeled “Other.”
- IGG0CLC6 inserts the new IPE into the index block left by IGG0CLC2. IGG0CLC6 writes the updated block to SYSCTLG, and ripples the effect of the change down the index chain, if necessary. The last block of the chain is left in the input/output buffers, but it is not written to SYSCTLG.
- IGG0CLC7 writes the last block of the updated index chain, then reads, updates, and re-writes blocks containing the index control entry (ICE) and volume index control entry (VICE). Resources are released and control passes back to the caller.

Traces such as the one just discussed are illustrated in the chapter “Diagnostic Aids” on page 60, as an aid in identifying the modules involved in any particular situation.

All of the catalog-management modules are reentrant. They use a common work space, initialized by IGC0002F and called WORKAREA. (WORKAREA is described on page 54.)

## Macros and Services Used

A small set of macros and services is used throughout the catalog-management routines. The prologue commentary for each module lists the specific ones used in that module; the complete set is:

- BLDL, the routine used by Phase I to read blocks of SYSCTLG, which is formatted like a partitioned data set (PDS) directory. It is accessed through entry point IECPBLDL, whose address is found in field CVTPBLDL of the Communication Vector Table (CVT). In Phase I catalog-management routines, this routine is used in a closed subroutine labeled “CALLBLDL”.
- IECPCNVT, a routine used to convert relative track addresses to absolute addresses. It is accessed through entry point IECPCNVT, whose address is found in field CVTPCNVT of the CVT. In the catalog management routines, this routine is used in the closed subroutine labeled “TOABSL”.
- IECPRLTV, a routine used to convert absolute track addresses to relative addresses. It is accessed through entry point IECPRLTV, whose address is found in field CVTPRLTV of the CVT. In the catalog-management routines, this routine is used in a closed subroutine labeled “TORLTV”.
- ENQ, a macro instruction used to control three resources: high-level name, volume index, and volume index control entry (VICE).
- DEQ, a macro instruction used to release resources obtained by the ENQ macro.
- EXCP, a macro instruction used for input/output operations in Phase II and Phase III modules. In the catalog management routines, this instruction is used in a closed subroutine labeled “IO”.

- EXIT, a SVC instruction used to return to the caller.
- XCTL, a macro instruction used to pass control from one catalog management module to another.
- GETMAIN, a macro instruction used to allocate main storage, used by catalog management to obtain work areas and input/output buffers.
- WAIT, a macro instruction that allows an input/output operation to complete before continuing processing.
- FREEMAIN, a macro instruction used to release main storage used for work areas by catalog management.
- SCRATCH, a macro instruction used to delete a data set.
- DSECT macros IEFUCBOB and CVT, which describe system control blocks.

### Character Dependency

The modules of catalog management require that the character set used at execution time be equivalent to that used at assembly time. The IBM-supplied version of catalog management assumes EBCDIC character representations. If a different character set is to be used during execution, the modules must be re-assembled. The instructions involved in this dependency are identified by label in the prologue commentary of each module.

### Resource Enqueuing

Figure 7 shows when the resources of catalog management are enqueued in relation to the modules. Three resources are used: high-level name, volume index, and volume index control entry (VICE). To prevent an interlock between two callers, the high-level name is *always* enqueued first, the volume index is enqueued second, and the VICE is enqueued last.

The conditions of enqueuing are determined from the request. If the volume index is to be modified, then the volume index must be enqueued exclusively. Otherwise, it can be shared. If a locate function is requested, then the high-level name can be shared. If a non-locate function is requested, the high-level name is enqueued exclusively to protect all lower-level indexes under it.

The major name for enqueuing is always "SYSCTLG". The minor name is the high-level name with the UCB address appended to it, "SYSCTLG" with the UCB address appended to it, or zeros with the UCB address appended to it.

### Register Usage

With the exception of IGC0002H and IGG0CLF2, the catalog-management modules use a common set of registers. Module IGC0002F initializes these registers, and their contents remain throughout. Contents of registers not described are considered destroyed.

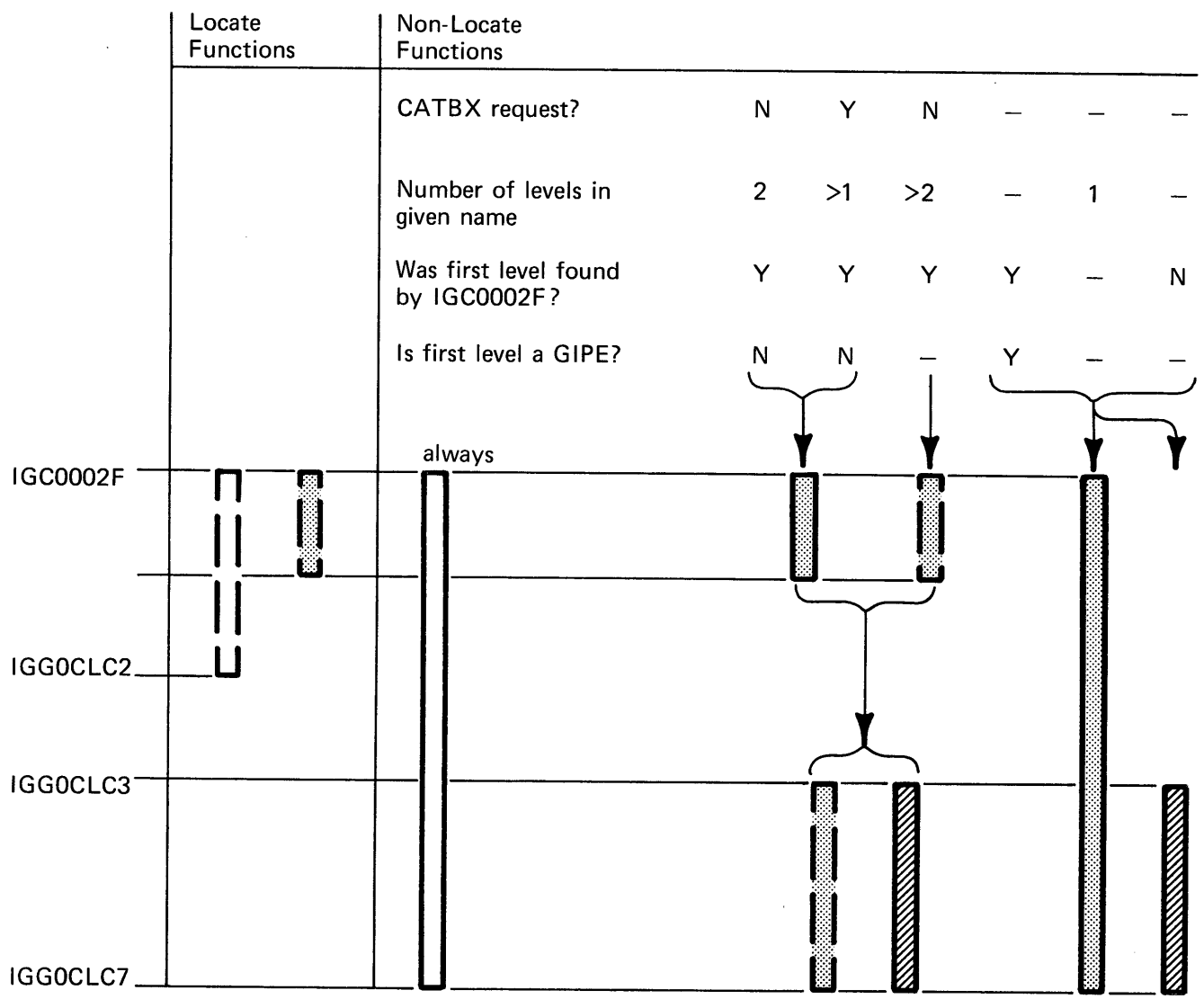
The register contents are:

- 4 Base register for the module
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BAL instructions
- 14 Linkage register for BAL instructions

### Module Descriptions

Each of the modules of catalog management is described in this section. The flowcharts are organized into subcharts; each subchart shows one function of the module. Supporting text for the module appears beside the subcharts. The labels on the flowchart are those used in the assembly listing.





Notes: "-" indicates "doesn't matter"

- shared enqueue
- exclusive enqueue
- High-level name
- Volume Index
- VICE

Order of enqueuing: 1. High-level Name  
 2. Volume Index  
 3. VICE (when applicable)

Missing combinations are error conditions

**Figure 7. Resource Enqueuing**

## IGC0002F: Initialization

IGC0002F is the entry point for the catalog-management routines, which are invoked by an SVC 26.

### Registers

#### On Entry:

- 1 Address of caller's parameter list (CAMLST)
- 3 Address of CVT
- 4 Address of TCB
- 5 Address of SVRB

#### On Exit:

- 2 Address of UCB
- 4 Base register for this module
- 5 Pointer to SVRB extension
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 9 Address of CVT
- 12 Linkage register for BAL instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BAL instructions

### Functions

The area WORKAREA is the common workspace and communications area for all catalog-management modules. When a locate function is requested, WORKAREA is built over the caller's 265-byte area, and a second area (called BLDLAREA) is obtained by GETMAIN. BLDLAREA is used with the routine BLDL.

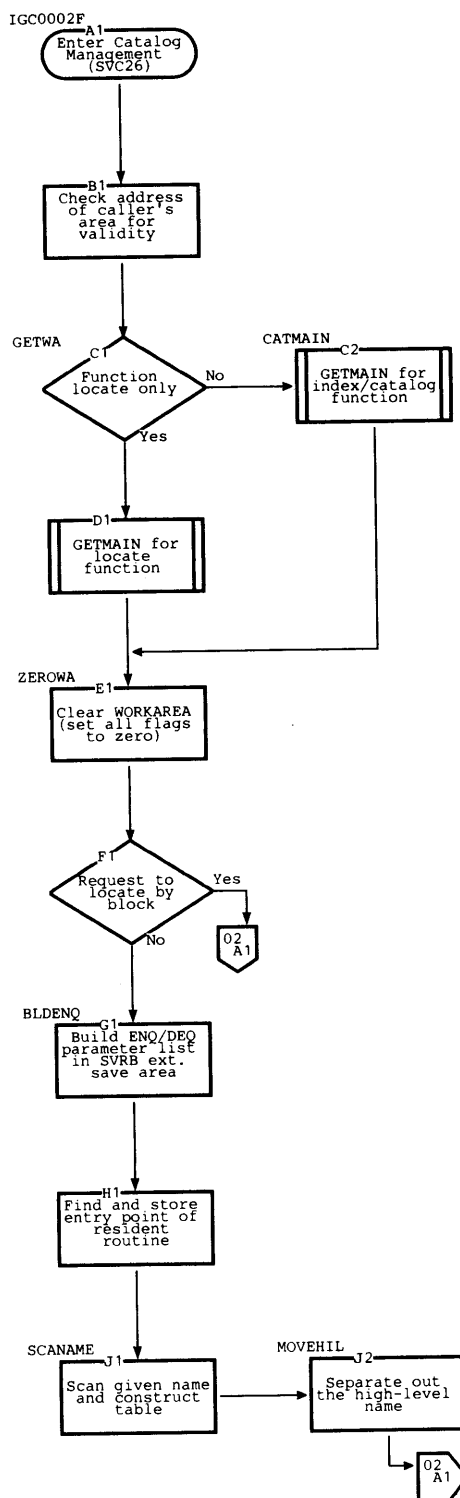
When a non-locate function is requested, a larger area is obtained by GETMAIN for WORKAREA. Part of this area is used for BLDLAREA during execution of modules IGC0002F, IGG0CLC1, and IGG0CLC2. This portion of WORKAREA is redefined for input/output buffers thereafter.

The first 256 bytes of WORKAREA are set to zero, which initializes all switches and flags. Supervisor addresses and the data set name go into WORKAREA, and the data set name is decomposed into its components. BLDLAREA is initialized for use as input/output buffers.

The UCB table is searched for device information for the given CVOL. GETMAIN is used to allocate space for a DCB and a DEB, and IGC0002H is used to open the CVOL (via an SVC 28).

**Note:** The OPEN macro is not used to open a CVOL. IGC0002H constructs a modified DCB/DEB for use by catalog management; IGC0002H is described on page 42. No CLOSE macro is issued to close a CVOL. FREEMAIN simply releases the main storage that is used for the modified DCB/DEB.

## Flowchart FC1. IGC0002F: Initialization (Part 1 of 2)



The first component of the data set name is used as the search parameter for BLDL. Searching begins with the first block of SYSCTLG. If BLDL returns a CVOL pointer entry (and the request is not LNKX or DRPX), the current CVOL is closed and control goes back to label UCBSRCH (block A1).

### Internal Subroutines

None.

### Exits

Control passes via an XCTL macro instruction to:

- IGG0CLC1 if requested function is BLDA or LNKX, or if the high-level name is an alias.
- IGG0CLC7 for an error condition.
- IGG0CLC2 otherwise.

Control passes via an SVC 28 to IGC0002H to open the catalog, and returns to this module.

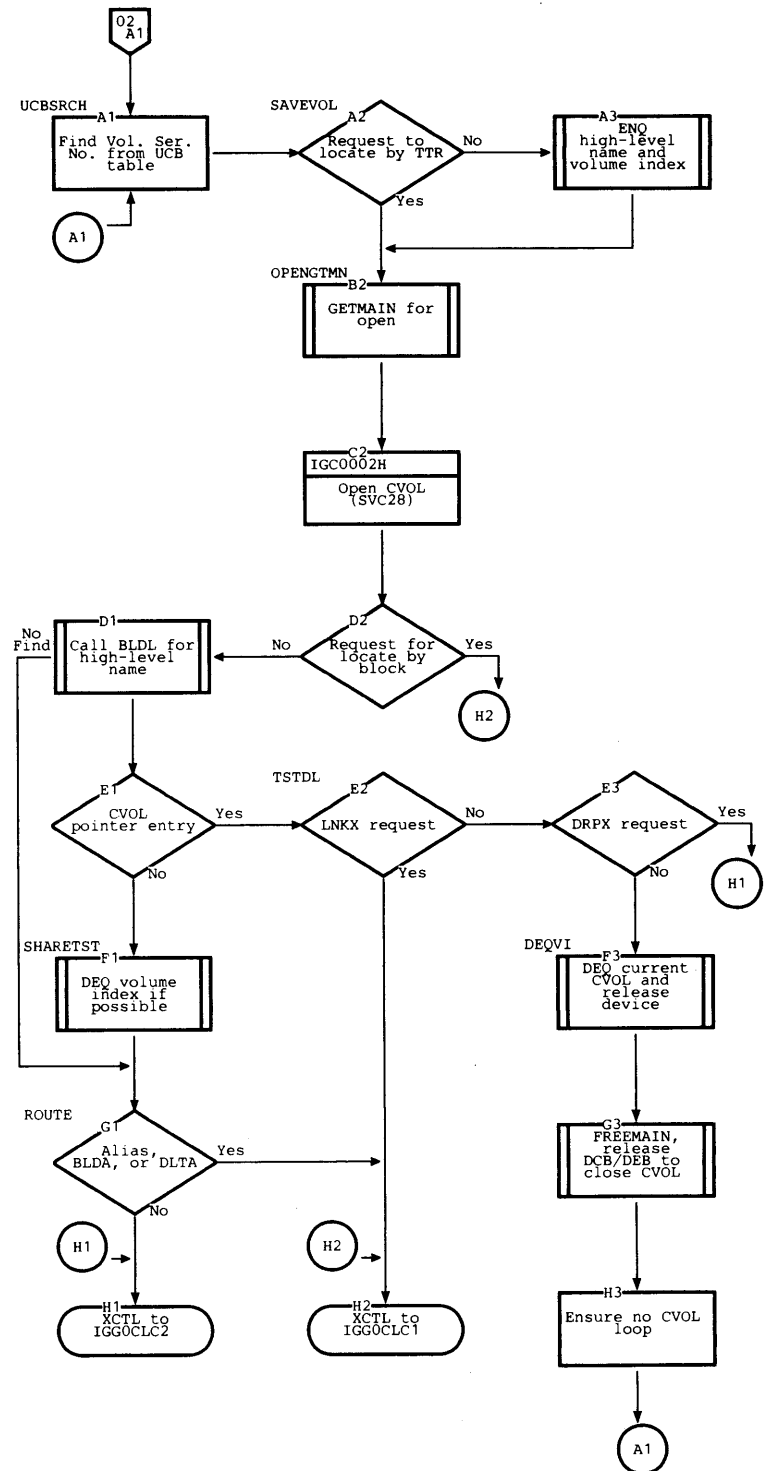
### Error Conditions

Code	Reason
4	CVOL not mounted or does not contain SYSCTLG.
20	Syntax error in data set name.
24	Permanent input/output error.
28	Bad relative track address for SYSCTLG.
32	Bad address for caller's area.

### References

Item	Page number
IGG0CLC1	28
IGG0CLC2	30
IGG0CLC7	40
IGC0002H	42
SYSCTLG Entries	47
CAMLSTD	53
WORKAREA	54
BLDLAREA	58
Method of Operation Diagram MO2	16
CVT, TCB, SVRB, DCB, DEB, and UCB are described in OS Control Blocks, Order Number GC28-6628.	

### Flowchart FC1. IGC0002F: Initialization (Part 2 of 2)



## IGG0CLC1: Relative GDG and Alias

IGG0CLC1 is the entry point.

Control comes from:

- IGC0002F when the requested function is BLDA or LNKX, locate-by-block, or when an alias is found (except with a DLTA request).
- IGG0CLC2 when a relative GDG number is found in the data set name.

### Registers

- 2 Address of UCB
- 4 Base register for this module
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BAL instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BAL instructions

### Functions

When locate-by-block is requested, the block is read and returned to the caller.

When control comes from IGG0CLC2, control goes to label RELGDG for relative GDG processing.

If the requested function is BLDA or LNKX, the appropriate entry is constructed and control passes to IGG0CLC2 for routing to the update modules.

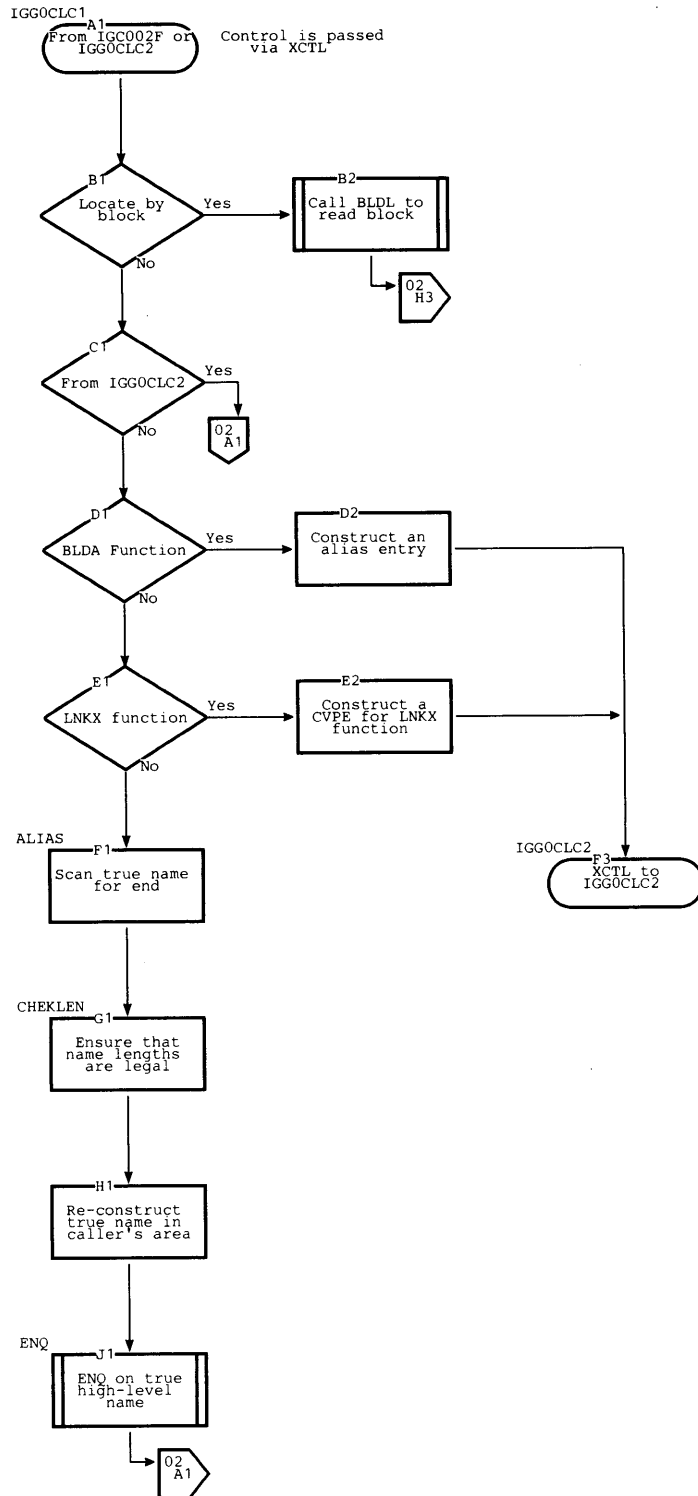
When an alias is discovered, the fully qualified name is reconstructed in the caller's name area, using the true name. The name table is then updated to reflect the change and the high-level name is re-enqueued.

Control comes from IGG0CLC2 when a relative GDG number is discovered in the data set name. This module determines the absolute GDG name for the data set. If the request is a locate function, either the volume list for the data set or a new absolute GDG name is returned to the caller. Otherwise, an error condition exists and IGG0CLC7 is invoked.

The generation number in absolute GDG names is complemented before the names are added to the generation index. Therefore, the most recent entry (the highest generation number) is the first entry in the index, the second most recent entry is the second entry in the index, etc.

When the relative GDG number is negative or zero, an absolute GDG name from the generation index is returned to the caller along with the corresponding volume list. Zero corresponds to the first entry, -1 corresponds to the second entry, etc.

## Flowchart FC2. IGG0CLC1: Relative GDG and Alias (Part 1 of 2)



When the relative GDG number is positive, a new absolute GDG name is created and returned to the caller. If the generation index is empty, this name is G000nV00 (where n is the relative number). If the generation index is not empty, the relative GDG number is added to the generation number of the first entry to create the new absolute GDG name.

**Internal Subroutines**

CALLBLDL calls BLDL routine via entry point IECPBBLDL.

**Exits**

Control returns to the caller after relative GDG processing (EXIT SVC instruction).

Control passes via an XCTL macro instruction to:

- IGG0CLC7 for error conditions.
- IGG0CLC2 otherwise.

**Error Conditions**

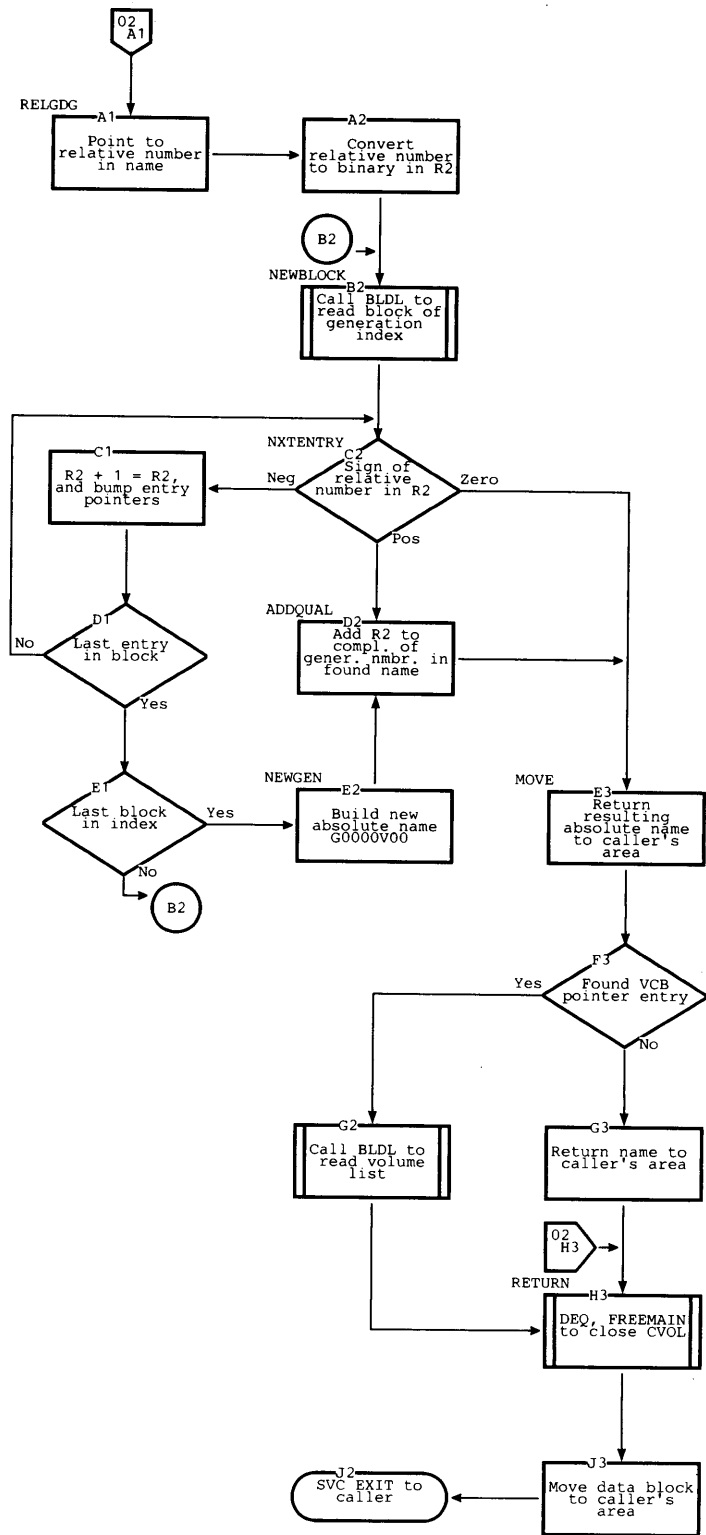
*Code Reason*

- 8 Name not found for locate function, or existing structure inconsistent with request for non-locate function.
- 20 Syntax error in data set name.
- 28 Permanent I/O error.

**References**

<i>Item</i>	<i>Page number</i>
IGC0002F	26
IGG0CLC2	30
IGG0CLC7	40
WORKAREA	54
Method of Operation Diagram MO2	16
Method of Operation Diagram MO3	18

**Flowchart FC2. IGG0CLC1: Relative GDG and Alias (Part 2 of 2)**



## IGG0CLC2: Locate

IGG0CLC2 is the entry point. Control comes from:

- IGG0CLC1 after resolving an alias or constructing an entry for BLDA or LNKX request.
- IGC0002F otherwise.

### Registers

- 2 Address of UCB
- 4 Base register for this module
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BALR instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BALR instructions

### Functions

This module completes the locate functions, or finds the last valid index level for a non-locate function. BLDL is used to search index levels successively. At each index level, one component of the data set name is used. When locate-by-name is requested, BLDL is used with each component of the data set name as the search parameter. When BLDL returns an index pointer entry (IPE), use it to determine the track address for the next search. The search by BLDL continues with the next component of the name.

When BLDL returns a data set pointer entry (DSPE) or volume control block pointer entry (VCBPE), the corresponding volume list is returned to the caller.

When the request is for a non-locate function, and BLDL fails to find the next level, the update process is initiated.

The last valid level of the existing index structure is saved to use while updating.

IGG0CLC2 contains skeletal channel programs that are used by the non-locate modules. These CCW chains are moved to BLDLAREA.

### Internal Subroutines

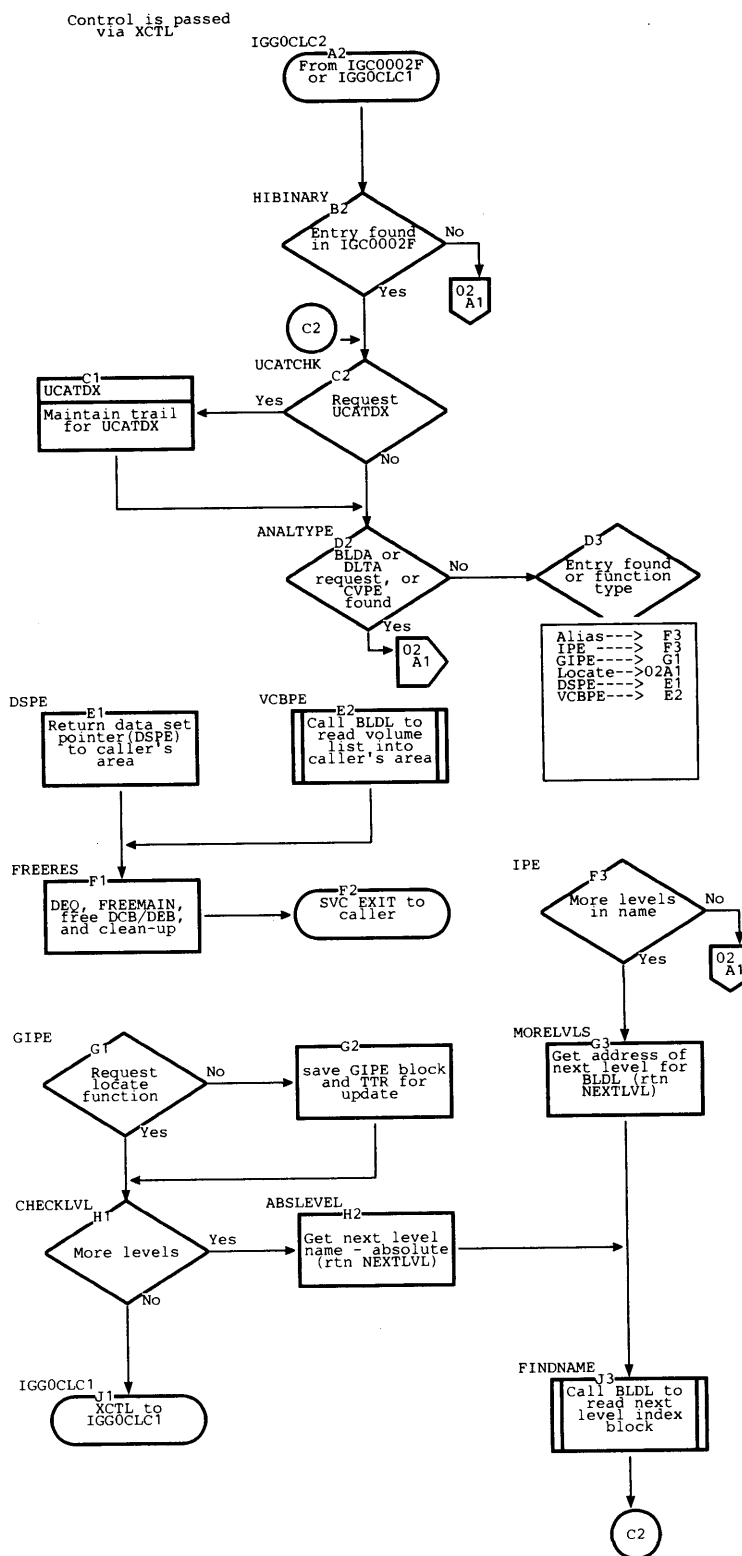
UCATDX maintains a TTR trail of blocks that can be deleted.

CALLBLDL calls BLDL to search for one name.

TORLTV converts an absolute address to a relative track address.

NEXTLVL gets the component of the data set name in order to search for the next level.

## Flowchart FC3. IGG0CLC2: Locate (Part 1 of 2)



### Exits

When the request is a locate function, control returns to the caller along with the volume list for the data set name (via EXIT SVC instruction).

When the request is for a non-locate function, control passes to:

- IGG0CLC1 for relative GDG number.
- IGG0CLC7 for an error condition.
- IGG0CLC3 otherwise.

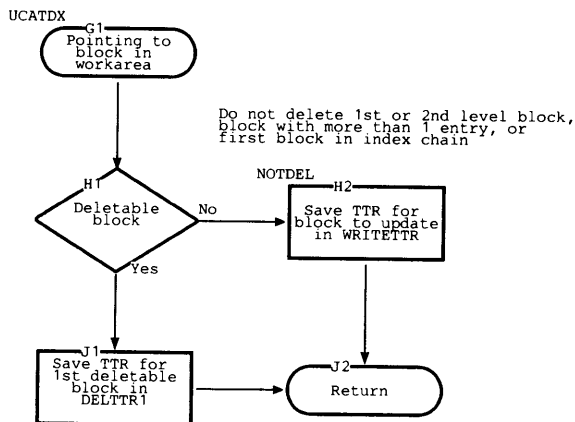
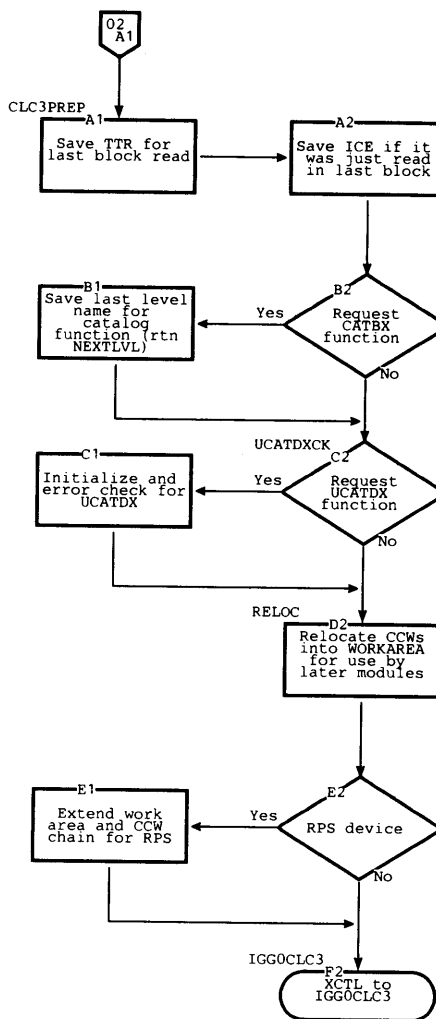
### Error Conditions

Code	Reason
8	name not found for locate request or existing structure inconsistent with non-locate request.
12	Last entry found was an IPE, CVPE, or alias with locate request.
16	Non-existent index levels specified.
20	Syntax error in data set name.
28	Permanent input/output error.

### References

Item	Page number
IGC0002F	26
IGG0CLC1	28
IGG0CLC3	32
IGG0CLC7	40
WORKAREA	54
BLDLAREA	58
Method of Operation Diagram MO2	16

**Flowchart FC3. IGG0CLC2: Locate (Part 2 of 2)**



## IGG0CLC3: Update Initialization and Entry Building

IGG0CLC3 is the entry point. Control comes from:

- IGG0CLC4 after constructing a DSPE for a CATBX request.
- IGG0CLC5 after writing a volume control block and constructing a VCBPE for a CATBX function.
- IGG0CLC2 otherwise.

### Registers

- 2 Address of UCB
- 4 Base register for this module
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BAL instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BAL instructions

### Functions

When entry is from IGG0CLC4 or IGG0CLC5, index levels for a CATBX request must be built. Control goes to label CATBX on the next subchart.

When control comes from IGG0CLC2, the index control entry (ICE), if not already present, and volume index control entry (VICE) are read. The request is checked against available space in SYSCTLG to ensure that there is space enough to make the required changes.

This module constructs new index levels for a BLDX function and constructs an index pointer entry for the new level to be added to the existing structure. When the requested function is DRPX or DLTA, the entry to be removed is named and IGG0CLC6 will delete it.

When CATBX is requested, IGG0CLC4 is called to construct the DSPE. Control returns to IGG0CLC3, where the required index levels are built and written into SYSCTLG. Each level results in an index pointer entry (IPE) that must be added to the next higher level. When an existing level is reached, control passes to IGG0CLC6.

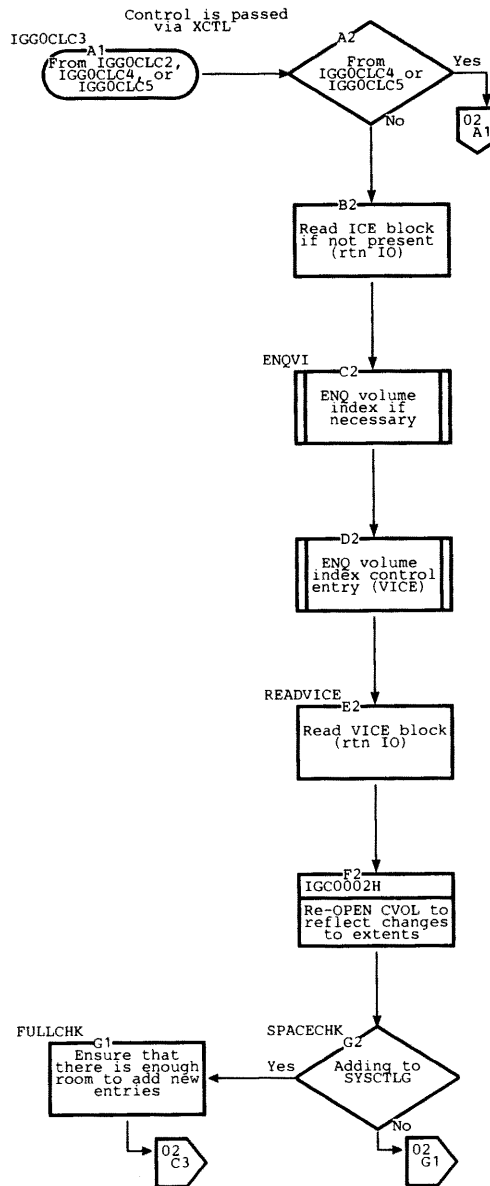
IGG0CLC3 routes the update request to the modules that perform the appropriate function.

### Internal Subroutines

MOVEVLV gets the component of the data set name for the current index level from the name table.

WRTSRCH writes a new block to SYSCTLG and searches for another available block.

Flowchart FC4. IGG0CLC3: Update Initialization and Entry Building (Part 1 of 2)





KEYICE constructs a new index block, with its ICE and key.

TOABSL converts a relative track address to an absolute track address.

TORLTV converts an absolute track address to a relative track address.

IO performs EXCP input/output. This subroutine invokes IGC0002H if a new extent of SYSCTLG is required.

**Exits**

Control passes via an XCTL macro instruction to:

- IGG0CLC4 when the requested function is CATBX, CAT, RECAT, or UNCAT.
- IGG0CLC5 when blocks of SYSCTLG need to be freed, or when new blocks have been written but the requested process has been aborted.
- IGG0CLC7 for error conditions.
- IGG0CLC6 otherwise.

Control passes via an SVC 28 to IGC0002H when a new extent of SYSCTLG is required, and returns to this module.

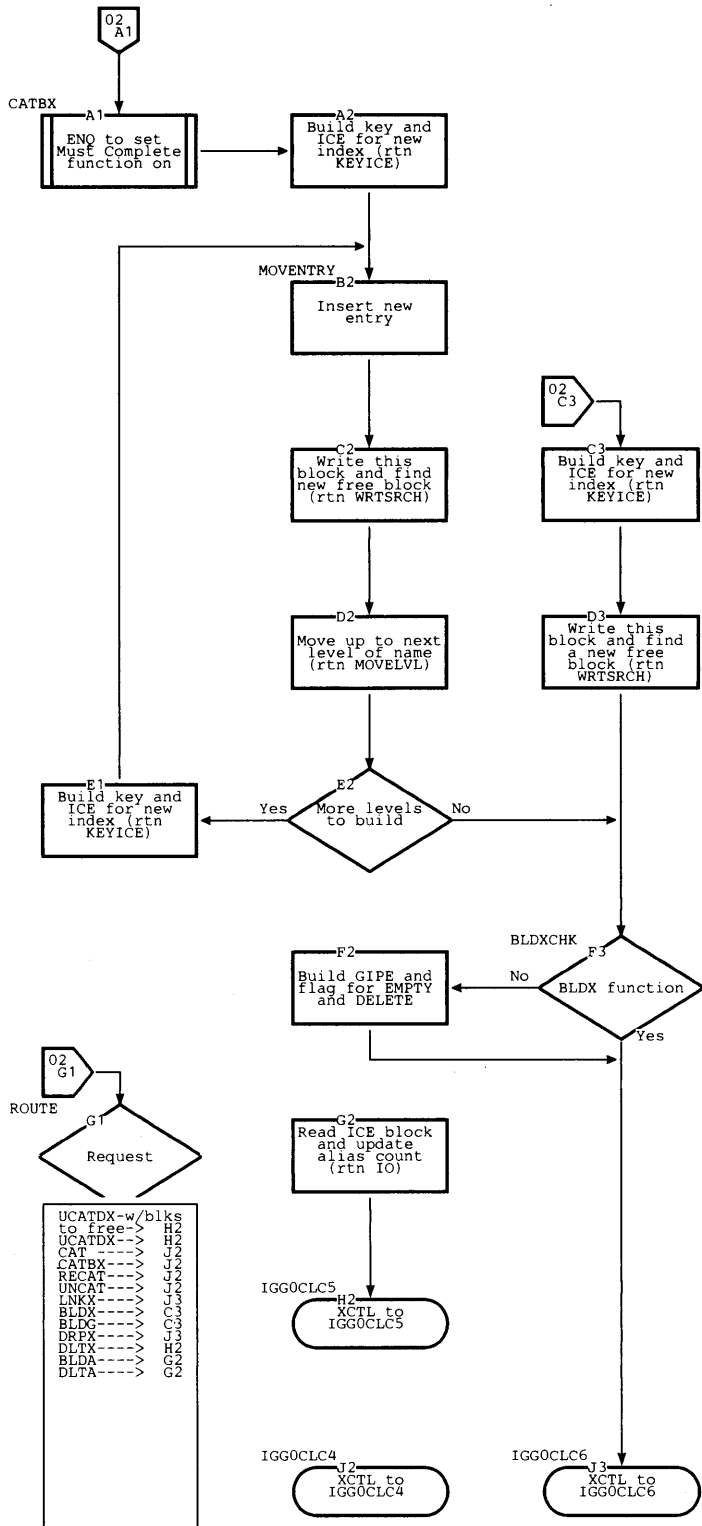
**Error Conditions**

- |             |  |
|-------------|--|
| <i>Code</i> | <i>Reason</i>  |
| 8           | Existing structure is inconsistent with the requested function.          |
| 12          | Attempt to delete a non-empty index level.                               |
| 20          | Not enough space available in SYSCTLG to perform the requested function. |
| 28          | Permanent input/output error.  |

**References**

<i>Item</i>	<i>Page number</i>
IGG0CLC2	30
IGG0CLC4	34
IGG0CLC5	36
IGG0CLC6	38
IGG0CLC7	40
IGC0002H	42
SYSCTLG Entries	47
WORKAREA	54
BLDLAREA	58
Method of Operation Diagram MO3	18
Method of Operation Diagram MO4	19

**Flowchart FC4. IGG0CLC3: Update Initialization and Entry Building (Part 2 of 2)**



## IGG0CLC4: Entry Building

IGG0CLC4 is the entry point. Control comes from IGG0CLC3 when the requested function is CAT, CATBX, RECAT, or UNCAT.

### Registers

- 2 Address of UCB
- 4 Base register for this module
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BAL instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BAL instructions

### Functions

If the requested function is RECAT or UNCAT, control passes to label ALTERN. If the request is for CAT or CATBX, control passes to label CATRTN.

This module constructs a new DSPE or VCBPE. When there are more than five volumes in the volume list, IGG0CLC5 is invoked to write volume control blocks. When there are five or fewer volumes, control passes back to IGG0CLC3 to construct an index level for the new DSPE.

If the data set name is not for a generation data group, control passes to label CULMINAT. Label CATGDG deals with cataloging functions to a generation index. The new member of a GDG is checked against existing members to see if this is a new version of an existing member.

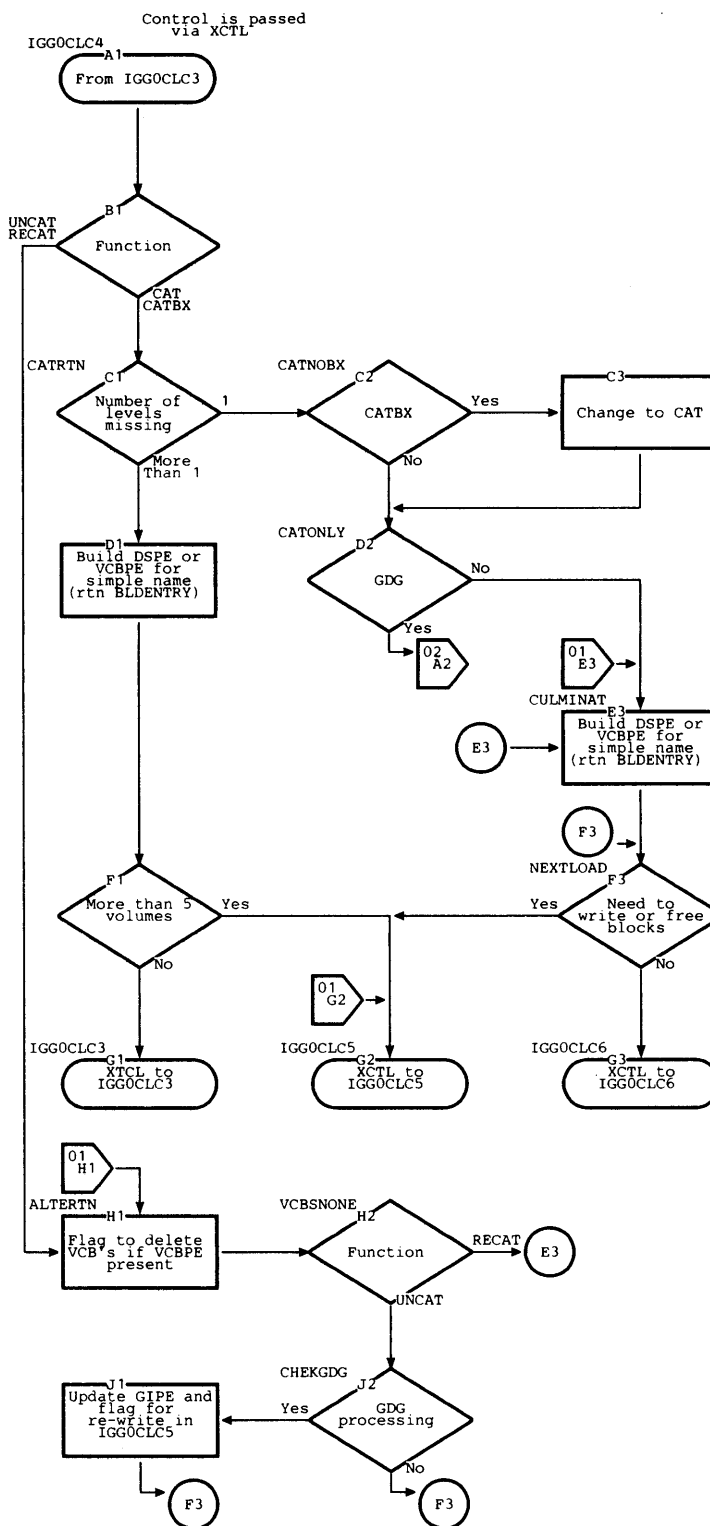
If the maximum number of entries that a generation index can hold is exceeded with this addition, the options for GDG are processed.

If EMPTY was specified, IGG0CLC5 will remove all entries from the generation index before adding the new entry. Otherwise, IGG0CLC5 will remove only the oldest entry before adding the new entry. This module (IGG0CLC4) flags what is to be done.

If DELETE was specified, then this module (IGG0CLC4) issues the SCRATCH macro instruction on every data set name that will be removed by IGG0CLC5. If DELETE was not specified, nothing is scratched.

The RECAT and UNCAT functions are processed by naming the old entry. IGG0CLC6 will delete the old entry when it gets control. For RECAT, a new entry is also constructed. IGG0CLC6 will add this new entry to the catalog.

Flowchart FC5. IGG0CLC4: Entry Building (Part 1 of 2)



### Internal Subroutines

TOABSL converts an absolute track address to a relative track address.

IO performs EXCP input/output operations.

GET reads a block from SYSCTLG into the input buffer of BLDLAREA.

SETUP points to the first and last entry in an index block.

INCR bumps the pointer to the next entry in an index block.

BLDENTRY constructs a data set pointer entry (DSPE) or a volume control block pointer entry (VCBPE).

SCRATCH performs SCRATCH macro instruction for one data set and its VCBs.

### Exits

Control is passed via an XCTL macro instruction to:

- IGG0CLC3 when CATBX is being performed.
- IGG0CLC5 when auxiliary reading or writing is required:
  - Volume control blocks (VCBs) need to be written.
  - VCBs or index blocks need to be freed.
  - The DELETE option of a GDG needs to be performed.
  - Updated GDG index blocks need to be rewritten.
- IGG0CLC7 for error conditions.
- IGG0CLC6 otherwise.

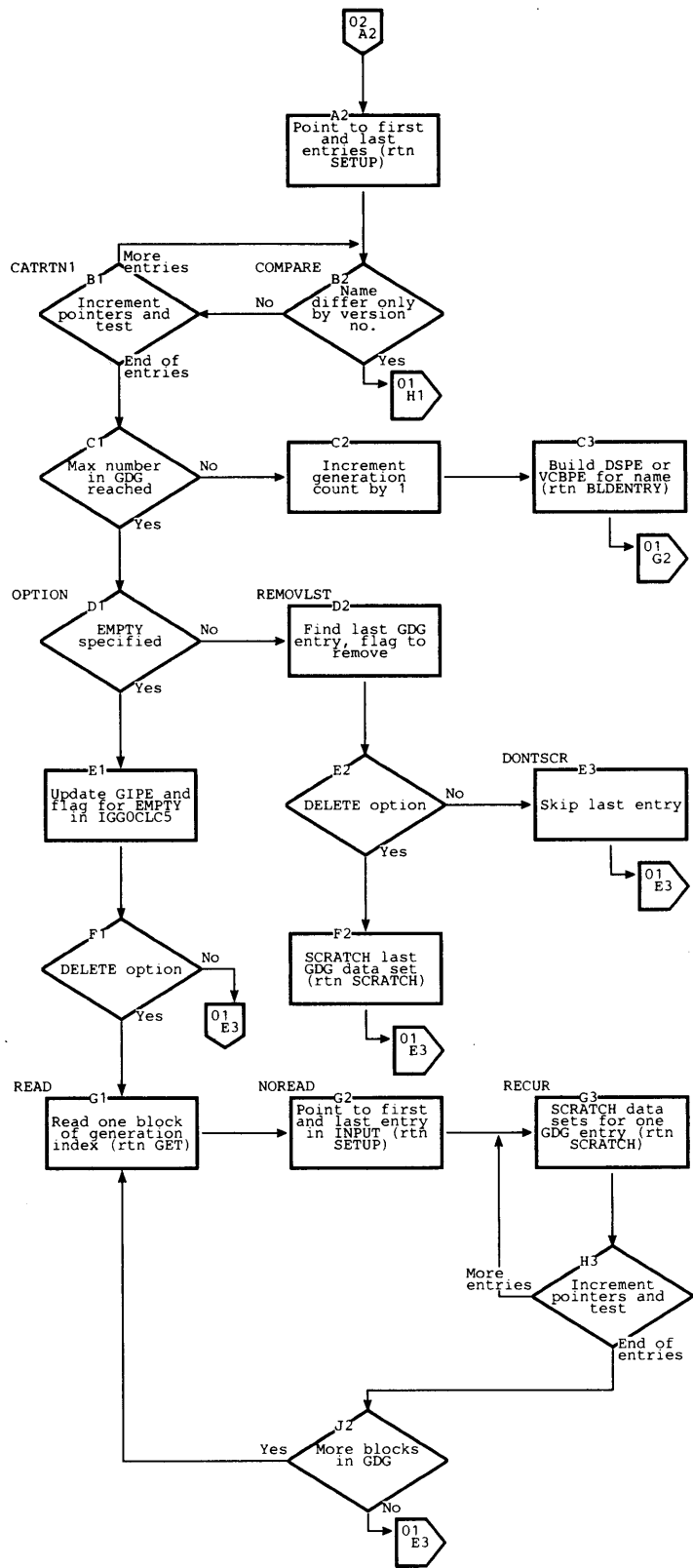
### Error Conditions

Code	Reason
8	Existing structure is inconsistent with requested function.
16	Non-existent index level required.
24	Improperly named GDG data set, or GDG data set to be added is older than existing GDG data sets.
28	Permanent input/output error.

### References

Item	Page number
IGG0CLC3	32
IGG0CLC5	36
IGG0CLC6	38
IGG0CLC7	40
SYSCTLG Entries	47
WORKAREA	54
Method of Operation Diagram MO3	18

### Flowchart FC5. IGG0CLC4: Entry Building (Part 2 of 2)



## IGG0CLC5: First Load of Update

IGG0CLC5 is the entry point. Control comes from IGG0CLC3 or IGG0CLC4 when blocks of the catalog need to be written or freed.

### Registers

- 4 Base register for this module
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BAL instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BAL instructions

### Functions

ENQ is reissued to set this task in the "Must Complete" mode and ensure that any changes to SYSCTLG will be completed.

This module consists of a series of tests for required functions. Each test calls the appropriate internal subroutine to perform one function if it is required.

Chains of volume control blocks (VCBs) and index blocks are freed if possible; that is, they are set to zeros and rewritten into SYSCTLG. They then have a key of zero, indicating that they are available for use.

If changes have been made to a generation index, the block containing the generation index pointer entry (GIPE) must be updated. Likewise, the last block of the generation index may need to be rewritten.

If a generation index reached its maximum number of entries in IGG0CLC4 and the EMPTY option was specified, that option is processed. IGG0CLC4 will have already processed the DELETE option.

If the generation index is full and the EMPTY option was not specified, the name with the lowest generation number (the oldest data set) is removed from the index.

An UCATDX request can result in unneeded index blocks. Such blocks are freed.

If a CATBX function is requested and the volume list contains more than five volumes, volume control blocks are constructed from that list and written to SYSCTLG.

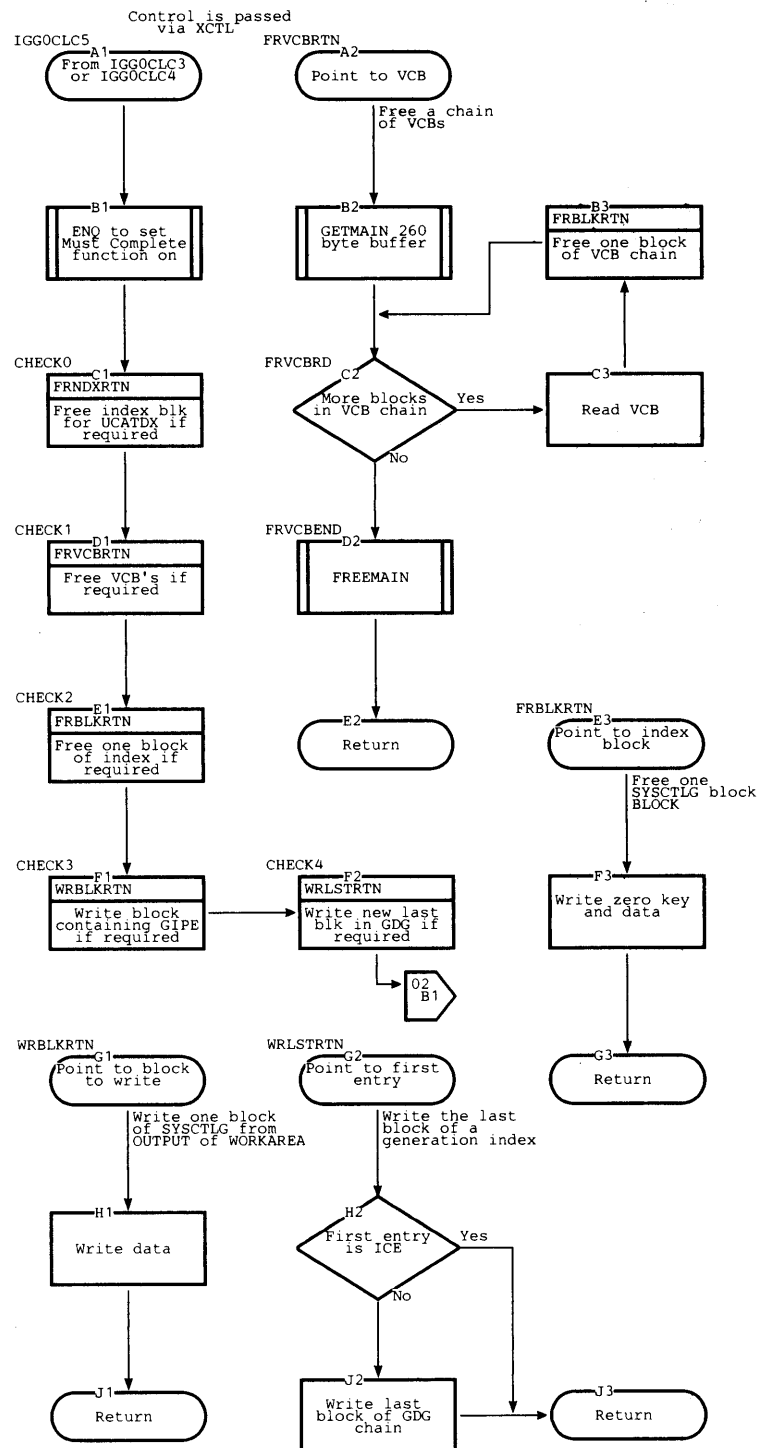
### Internal Subroutines

WRBLKRTN, WRLSTRTN, EMPTYRTN, FRNDXRTN, FRVCBRTN, FRBLKRTN, and BLVCBRTN are shown on the flowchart.

SETUP points to the first and last entry in an index block.

INCR increments the pointer to the next entry in an index block.

## Flowchart FC6. IGG0CLC5: First Load of Update (Part 1 of 2)



TOABSL converts a relative track address to an absolute track address.

TORLTV converts an absolute track address to a relative track address.

IO performs EXCP input/output operations. This subroutine invokes IGC0002H if a new extent is required.

### Exits

Control passes via an XCTL macro instruction to:

- IGG0CLC3 when the requested function is CATBX.
- IGG0CLC7 for error conditions.
- IGG0CLC6 otherwise.

Control passes via an SVC 28 to IGC0002H when a new extent of SYSCTLG is required, and returns to this module.

### Error Conditions

The only error condition detected in this module is code 8, which indicates that the existing structure is inconsistent with the requested function.

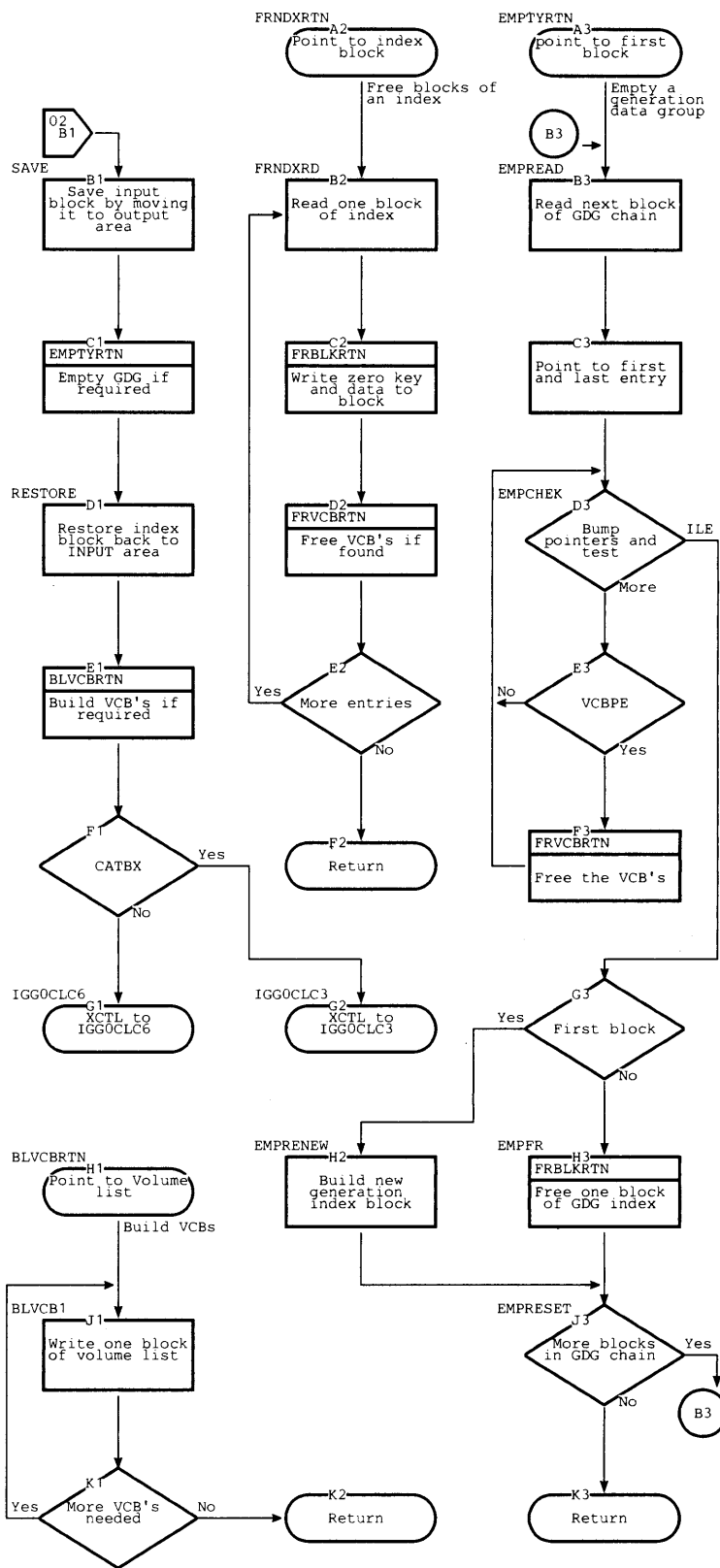
#### Code Reason

- 20 Not enough space available in SYSCTLG to perform the requested funtion.
- 28 Permanent input/output error.

### References

Item	Page number
IGG0CLC3	32
IGG0CLC4	34
IGG0CLC6	38
IGG0CLC7	40
IGC0002H	42
SYSCTLG Entries	47
WORKAREA	54
Method of Operation Diagram MO3	18
Method of Operation Diagram MO4	19

Flowchart FC6. IGG0CLC5: First Load of Update (Part 2 of 2)



## IGG0CLC6: Second Load of Update

IGG0CLC6 is the entry point. Control comes from:

- IGG0CLC4 when the requested function is CAT, UNCAT, RECAT, or CATBX.
- IGG0CLC3 or IGG0CLC5 otherwise.

### Registers

- 4 Base register for this module
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BAL instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BAL instructions

### Functions

This module adds or deletes an entry from a given index block, as set up by earlier phases, and ripples the change through the index chain as needed. Each entry is taken from the buffer INPUT and placed into the buffer OUTPUT until the collating sequence of the entry is equal to or greater than the name in the update request. If the request name is equal, that entry is skipped (delete function). If the request name is greater, the new entry is merged into OUTPUT (add function). Overflow entries become an add request for the next block in the chain.

Subroutines named GET and PUT are used for input/output. GET reads a block into INPUT, a field of WORKAREA, and initializes PUT. Entries are transferred from INPUT to OUTPUT, another field of WORKAREA. When all entries have been exhausted from INPUT, another block of the index is read from SYSCTLG.

When OUTPUT is full, a block is written to SYSCTLG from OUTPUT by the routine PUT. PUT checks all possibilities before writing the block and chooses the record of SYSCTLG that is most likely to result in contiguous blocks of one index. PUT tries to free any unneeded blocks; any unneeded block that PUT cannot free is later freed by GET.

### Internal Subroutines

GET, reads one block from an index in SYSCTLG.

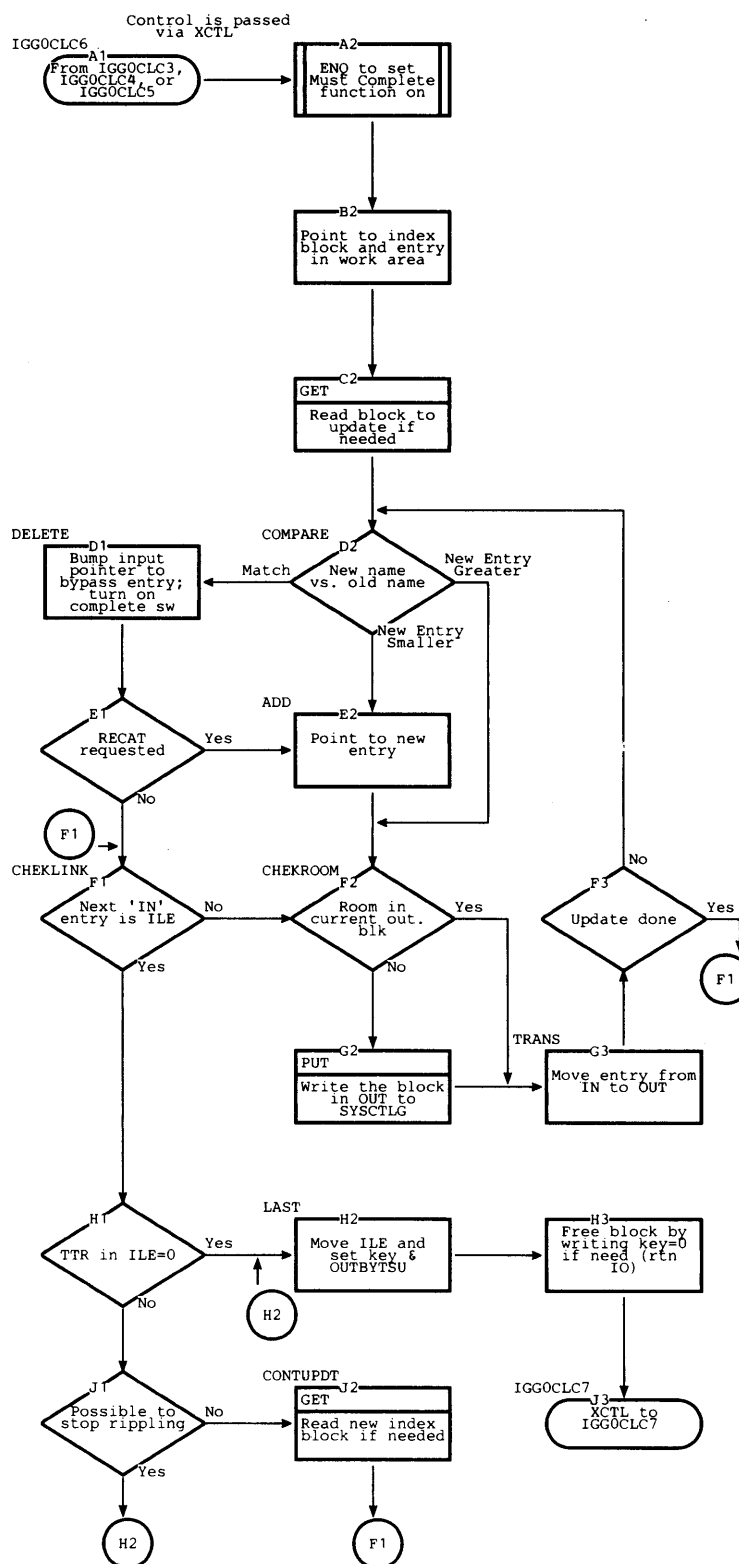
PUT, prepares and writes one block into an index in SYSCTLG.

TOABSL converts a relative track address to an absolute track address.

TORLTV converts an absolute track address to a relative track address.

IO performs EXCP input/output operations.

## Flowchart FC7. IGG0CLC6: Second Load of Update (Part 1 of 2)



**Exits**

Control is always passed to IGG0CLC7 via an XCTL macro instruction.

**Error Conditions**

The only exception code from this module is 28, which indicates that a permanent input/output error has occurred.

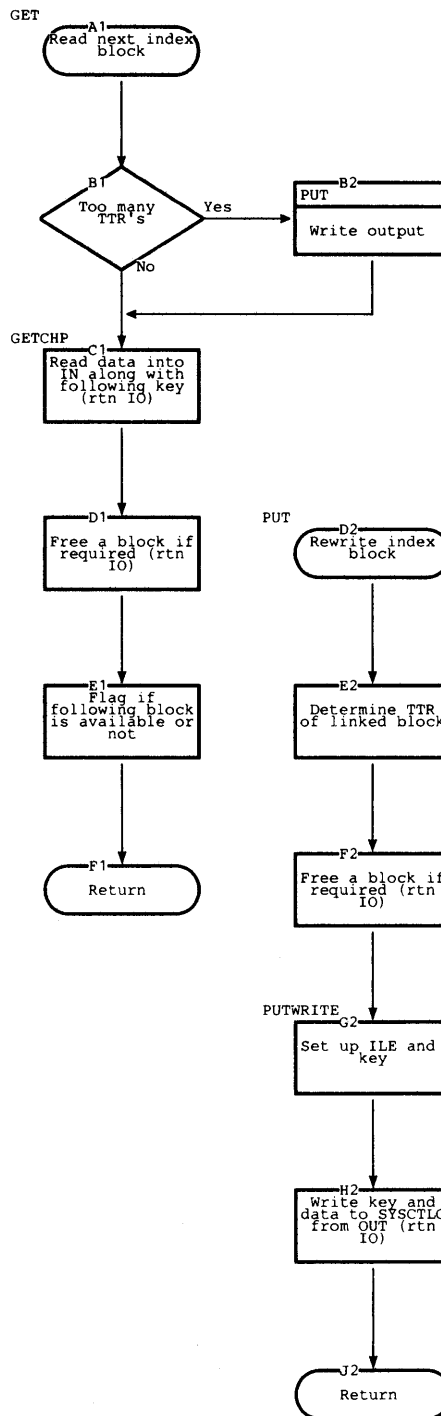
**References**

*Item*

IGG0CLC3	32
IGG0CLC4	34
IGG0CLC5	36
IGG0CLC7	40
SYSCTLG Entries	47
Method of Operation Diagram MO4	19

*Page number*

**Flowchart FC7. IGG0CLC6: Second Load of Update (Part 2 of 2)**



## IGG0CLC7: Third Load of Update and Error Handling

IGG0CLC7 is the entry point. Control normally comes from IGG0CLC6, but can come from any module of catalog management when an error condition is discovered.

### Registers

- 4 Base register for this module
- 5 Pointer to SVRB extension
- 6 Base register for WORKAREA DSECT
- 8 Base register for CAMLSTD DSECT
- 12 Linkage register for BAL instructions
- 13 Base register for BLDLAREA
- 14 Linkage register for BAL instructions

On exit to the caller, the registers (except registers 0, 1, and 15) are restored by the supervisor.

Register 15 contains the exceptional return code. Registers 0 and 1 contain additional information that specifies the type of error encountered.

### Functions

IGG0CLC7 completes the update process. The last block of an updated index is written to SYSCTLG.

The block containing the index control entry (ICE) is read, and the ICE is updated to reflect changes to the index. This block is rewritten into SYSCTLG.

The block containing the volume index control entry (VICE) is read, and the VICE is updated to reflect changes to SYSCTLG. This block is rewritten into SYSCTLG.

Tests are made before rewriting any block. If the block is both the last block of an index and the block containing the ICE, or the block containing the VICE, it is rewritten only once.

If an error is discovered, pertinent information is gathered into an environment record and written to SYSCTLG. The exceptional return code is set and all resources are freed. Control returns to the caller of catalog management via an EXIT SVC instruction.

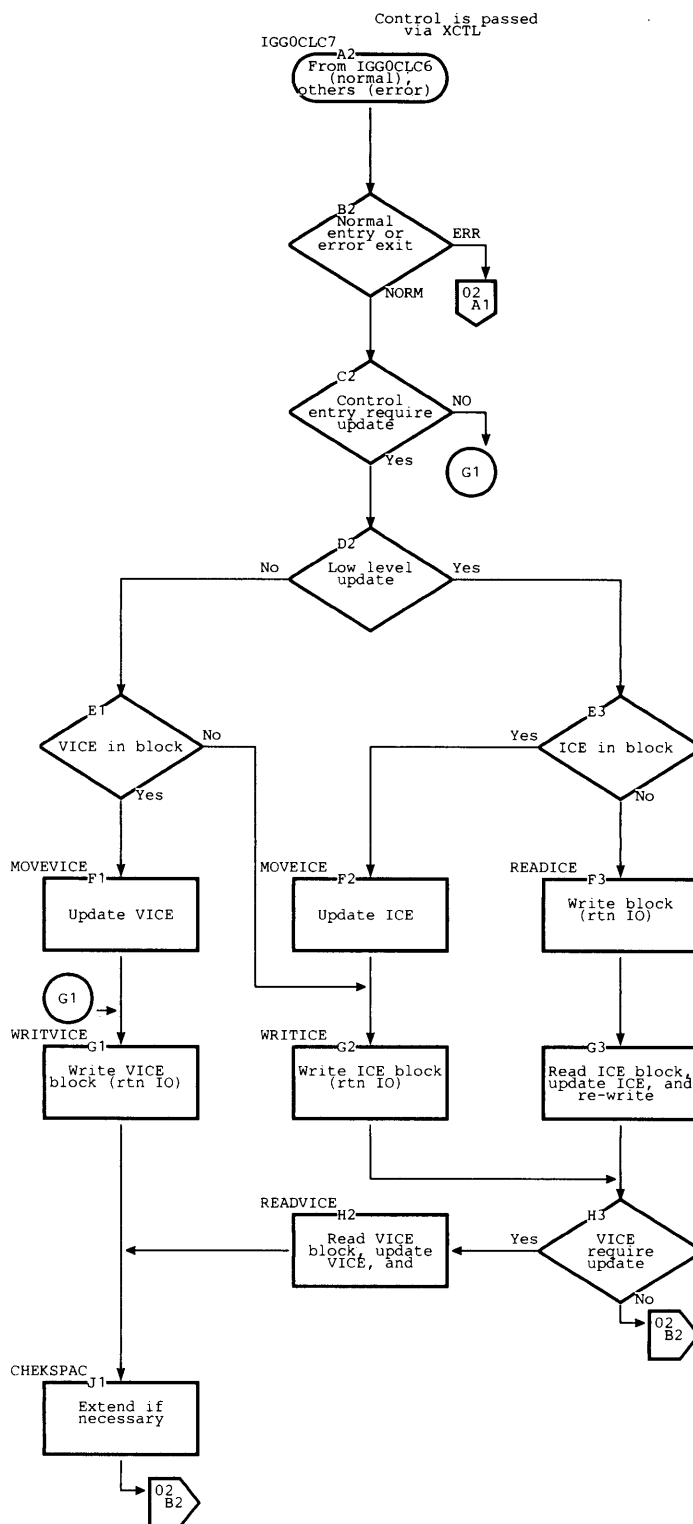
### Internal Subroutines

GETMAINB gets main storage for work space.

TOABSL converts a relative track address to an absolute track address.

TORLTV converts an absolute track address to a relative track address.

## Flowchart FC8. IGG0CLC7: Third Load of Update and Error Handling (Part 1 of 2)





IO performs EXCP input/output operations. This subroutine invokes IGC0002H if a new extent of SYSCTLG is required.

**Error Conditions**

This module returns any catalog management module exception code to the caller. This exception code is passed to IGG0CLC7 in WORKAREA.

The only exception code generated by this module is 28, which indicates that a permanent input/output error has occurred.

**Exits**

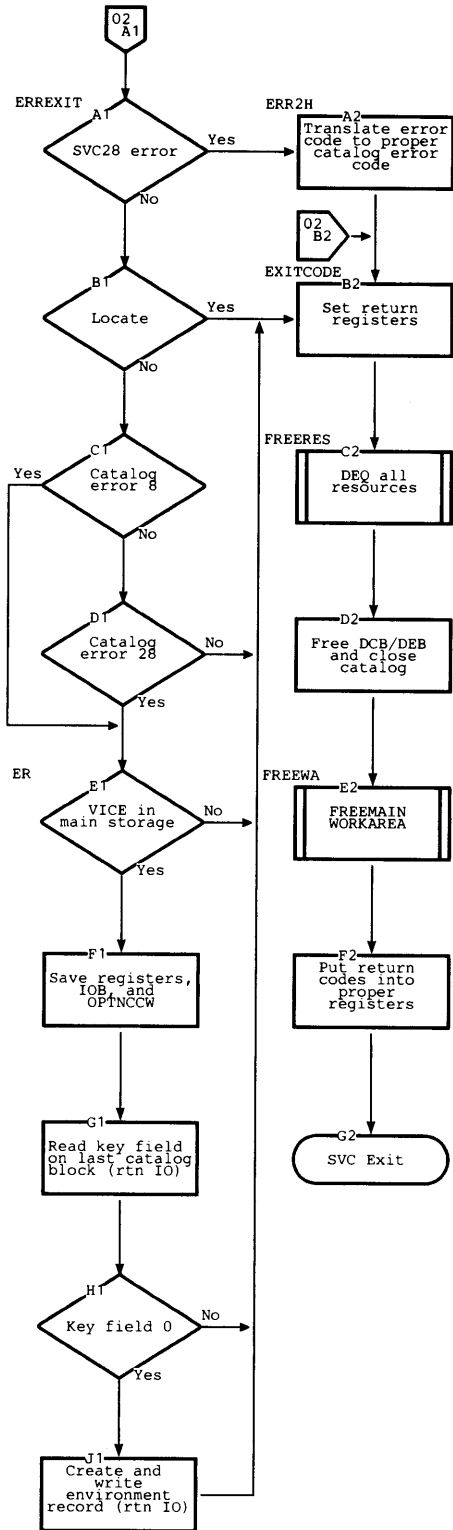
IGC0002H may be invoked via an SVC 28 when a new extent of SYSCTLG is required. Control returns to this module when one has been located.

Control always returns to the caller of catalog management via an EXIT SVC instruction.

**References**

<i>Item</i>	<i>Page number</i>
IGG0CLC6	38
IGC0002H	42
VICE	52
ICE	50
Environment Record	52
Method of Operation Diagram MO4	19

**Flowchart FC8. IGG0CLC7: Third Load of Update and Error Handling (Part 2 of 2)**



## IGC0002H: SYSCTLG Open/Extend

IGC0002H is the entry point. Control comes via an SVC 28 from:

- IGC0002F or IGG0CLC3 to open a CVOL.
- IGG0CLC3, IGG0CLC5, or IGG0CLC7 to extend SYSCTLG.
- IGG0553E after-extending SYSCTLG.

### Registers

On Entry for Open:

- 0 Zero
- 1 Address of UCB table for CVOL
- 15 Address of area in which to build DCB/DEB chain

On Entry for Extend:

- 0 Address of DCB for SYSCTLG

On Entry After Extending:

- 0 Negative number
- 4 Address of Extend work area
- 9 Address of DCB for SYSCTLG
- 10 TTR for new extent of SYSCTLG
- 11 Address of UCB

On Exit:

- 1 Address of DCB/DEB chain

### Functions

When this module is entered to open a CVOL, a data control block (DCB) and a data extent block (DEB) are built in the work area provided by IGC0002F. If the catalog is new, IGG0CLF2 is invoked to format it.

**Note:** The DCB/DEB constructed by this module is a modification of that described in *OS System Control Blocks*. See Figure 8.

When this module is entered for another extent of SYSCTLG, a test is made to see if another extent already exists. If so, **WORKAREA** is modified accordingly, and control returns to the caller.

When another extent does not exist, the main storage for the previous DCB/DEB is released and a new area is obtained with **GETMAIN**. **IGG0553A** is invoked to allocate a new extent, and a new DCB/DEB is built into the new area (the DCB/DEB is re-opened).

The DCB/DEB built by this module differs from that described in *OS System Control Blocks*. These two blocks are merged together; that is, they overlap in the same area of main storage, as shown in Figure 8. Main storage for the DCB/DEB is set to zeros before building; then only the fields that are shown are filled in. The first 20 bytes of the DCB do not exist (shown in dotted line), so the area starts with offset 20 of the

## Flowchart FC9. IGC0002H: SYSCTLG Open/Extend (Part 1 of 2)

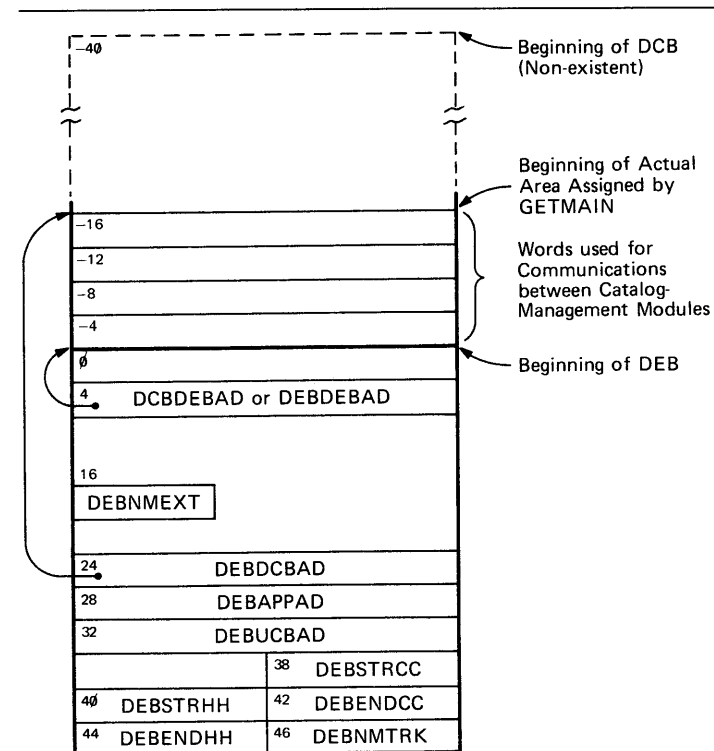
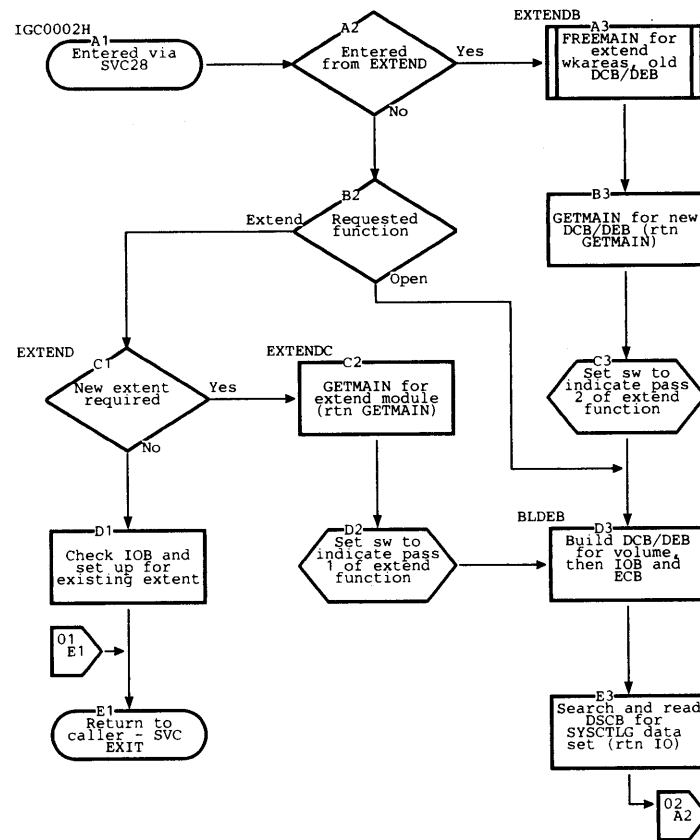


Figure 8. DCB/DEB Built by IGC0002H

DCB. The DEB overlays the DCB at offset 40. The fields that are named are described in *OS System Control Blocks*.

**Internal Subroutines**

GETMAIN gets main storage for DCB/DEB.

IO performs EXCP input/output operations.

**Exits**

Control returns to the caller via an EXIT SVC instruction when IGC0002H completes its function.

Control returns to the caller via an EXIT SVC instruction for an error condition.

Control passes via an XCTL macro instruction to:

- IGG0553A when another extent is required. Control returns via an XCTL macro instruction to the entry point, IGC0002H (block A1 of part A).
- IGG0CLF2 when the catalog or a new extent needs to be formatted. Control returns directly to the caller.

**Error Conditions**

*Code Reason*

- 4 No extents are allocated.
- 8 No more extents are possible.
- 12 Permanent input/output error.

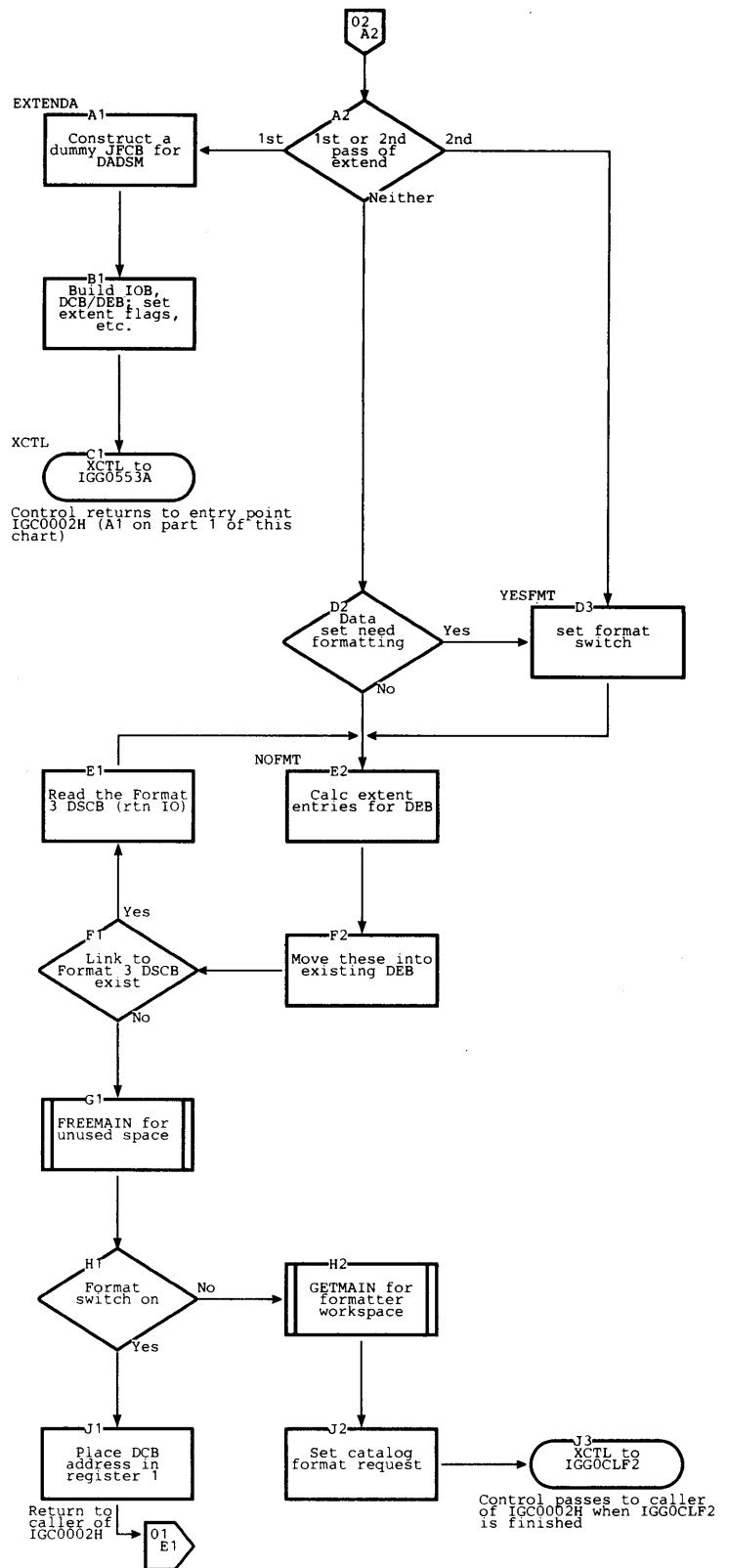
IGG0553A through IGG0553E are invoked by this module and can also return exception codes. See the publication for those modules (listed below) for details.

**References**

Item	Page number
IGC0002F	26
IGG0CLC3	32
IGG0CLC4	34
IGG0CLC5	36
IGG0CLF2	44

IGG0553A through IGG0553E are described in *OS DADSM Logic*, Order Number GY28-6607.  
 DCB and DEB are described in *OS System Control Blocks*, Order Number GC28-6628.

**Flowchart FC9. IGC0002H: SYSCTLG Open/Extend (Part 2 of 2)**



## IGG0CLF2: SYSCTLG and BPAM Directory Formatter

IGG0CLF2 is the entry point. Control comes from IGC0002H.

### Registers

- 0 Number of blocks to be written into a BPAM directory (zero when formatting SYSCTLG)
- 1 Address of DCB for this data set
- 2 Number of blocks per track for this device
- 3 Number of bytes in work area passed to IGG0CLF2
- 4 address of work area
- 5 Data management count decrement value
- 6 Starting relative track address (TTR) when formatting SYSCTLG

### Functions

The data set is formatted into 256-byte blocks with 8-byte keys. When formatting a BPAM directory, the module writes one block at a time. When formatting SYSCTLG, the module writes an entire track of blocks each time.

If this extent is being formatted during an open CVOL request, then this is the first extent of SYSCTLG. The first block is initialized by writing a volume index control entry into it.

If formatting is not being done during an open request, then this is an extent of an already existing SYSCTLG. Information to update the volume index control entry is placed into WORKAREA.

The work area passed to IGG0CLF2 is freed before exit.

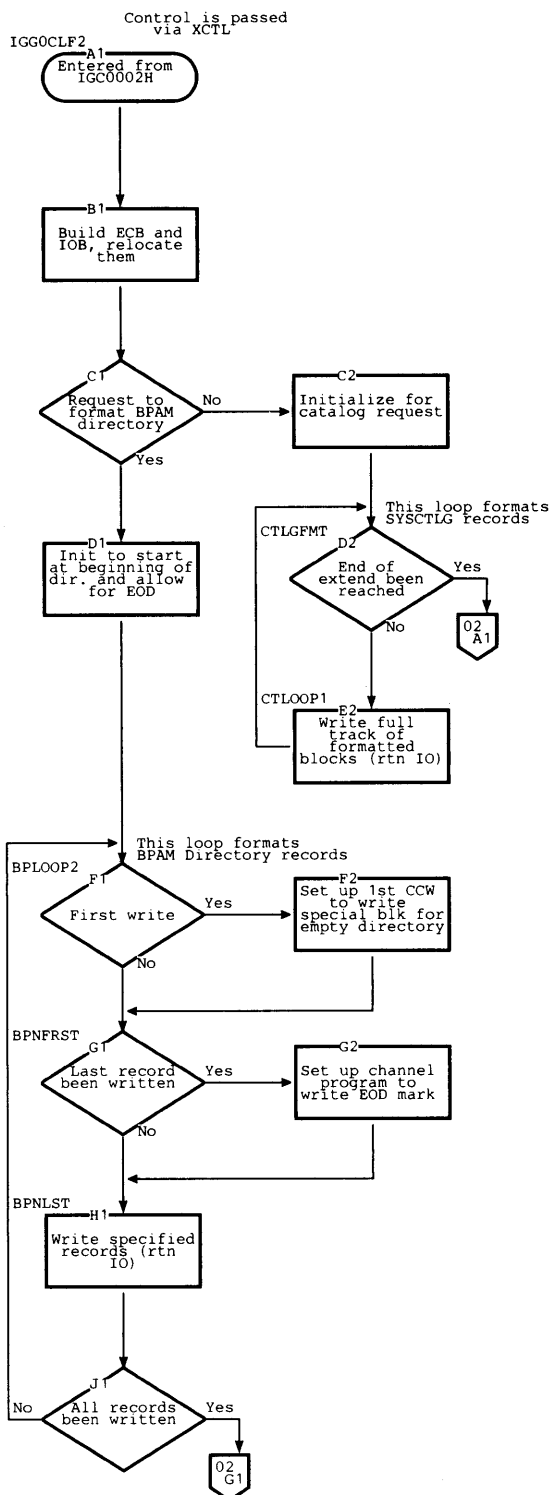
### Internal Subroutines

CNVT converts a relative track address to an absolute track address.

IO performs EXCP input/output operations.

RELOC builds channel programs for input/output.

## Flowchart FC10. IGG0CLF2: SYSCTLG and BPAM Directory Formatter (Part 1 of 2)



**Error Conditions**

IGG0CLF2 returns one exception code, 12, which indicates that an input/output error has occurred. The caller of catalog management never sees this code.

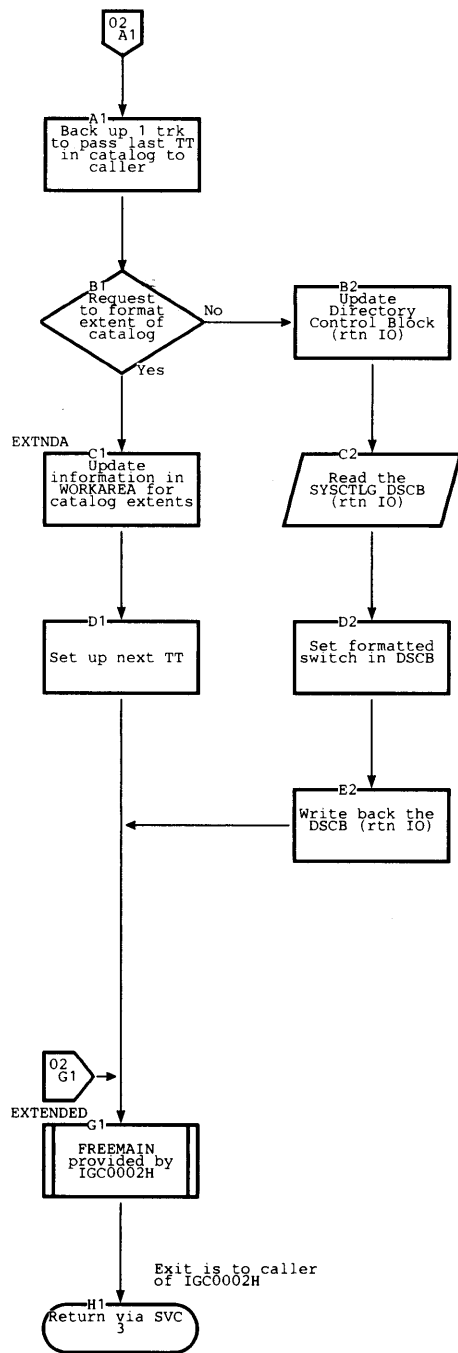
**Exits**

Control is passed to the caller of IGC0002H via an EXIT SVC instruction.

**References**

<i>Item</i>	<i>Page number</i>
IGC0002H	42
VICE	52
WORKAREA	54

**Flowchart FC10. IGG0CLCF2: SYSCTLG and BPAM Directory Formatter (Part 2 of 2)**



## Microfiche Directory

This chapter helps you relate the discussions in this book to the assembly listings found on microfiche. The rightmost columns of this directory contain page number references to Method of Operation diagrams and/or Flowcharts where more information can be found. The microfiche are found under the component name "Data Management."

<i>Entry Point Name</i>	<i>Module/ Microfiche Name</i>	<i>Description</i>	<i>Method of Operation Diagram</i>	<i>Flowchart</i>
IGC0002F	IGC0002F	Initialization for catalog management routines	MO2 (16)	FC1 (26)
IGG0CLC1	IGG0CLC1	Relative GDG and Alias processing	MO2 (16)	FC2 (28)
IGG0CLC2	IGG0CLC2	Locate processing	MO2 (16)	FC3 (30)
IGG0CLC3	IGG0CLC3	Update initialization and entry building	MO3 (18) MO4 (19)	FC4 (32)
IGG0CLC4	IGG0CLC4	Entry Building	MO3 (18)	FC5 (34)
IGG0CLC5	IGG0CLC5	First load of update	MO4 (19)	FC6 (36)
IGG0CLC6	IGG0CLC6	Second load of update	MO4 (19)	FC7 (38)
IGG0CLC7	IGG0CLC7	Third load of update, exit processing, and error handling	MO4 (19)	FC8 (40)
IGC0002H	IGC0002H	SYSCTLG open or extend		FC9 (42)
IGG0CLF2	IGG0CLF2	SYSCTLG and BPAM directory formatter		FC10 (44)

## Data Areas

The data areas and record formats in this chapter are described in five columns, which are interpreted as follows:

### *Offset*

The numeric address of the field relative to the beginning of the area. The first number is the offset in decimal, followed (in parentheses) by the hexadecimal equivalent.

### *Bytes and Alignment*

The size (number of bytes) of the field and its alignment relative to the full-word boundary.

Examples:

- 4        A 4-byte field beginning on a word boundary.
- .. 3     A 3-byte field beginning on a halfword boundary and running into the next word.
- ... 2    A 2-byte field beginning at the low-order byte of a word and running into the next word.

### *Name and Content*

A name that identifies the field. This name appears as a label in the assembly listings for the catalog-management routines.

This column is also used to show the contents of the field or the bit settings of flag fields (the state of bits in a byte). When the column is used to show the state of the bits (0 or 1) in a flag byte, it is shown as follows:

- .... ....    The 8 bit positions (0-7) in a byte. For ease of scanning, the high-order (left-most) 4 bits are separated from the low-order 4 bits.
- x... ....    A reference to bit 0.
- 1... ....    Bit 0 is on.
- 0... ....    Bit 0 is off.
- .... ..xx    A reference to bits 6 and 7.

Bit settings that are significant are shown and described. Bit settings that are not presently shown are understood to be reserved bits.

### *Field Description and Meaning*

The use of the field.

## **SYSCTLG Entry Formats**

This section describes the formats of the entries of SYSCTLG, along with the symbolic labels that are used to refer to their fields. The entries are arranged alphabetically.

Except for the volume control block (VCB), SYSCTLG entries have a similar format. These entries share a common definition for the first 12 bytes. The shared names are:

ENAME	ETTR	ETYPE
(8 bytes)	(3 bytes)	(1 byte)

Individually named fields follow either ETTR or ETYPE.

The entries in a SYSCTLG block begin in the third byte of the block. The first halfword of the block contains the binary number of the bytes that are used in this block, including the halfword count field.

**Alias Entry (AE)**

An alias entry defines an alternate name for the high-level qualifier of a data set name.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME	Name; contains the alias of the high-level index whose relative track address is found at offset 8 of this entry.
8 (8)	3	ETTR	Address; contains the relative track address (TTR) of the first block of the index named at offset 12 of this entry.
11 (B)	... 1	ETYPE X'04'	Type; indicates that this is an alias entry; also that four halfwords follow in the remainder of the entry.
12 (C)	8	ETRUEN	True name; contains the name of the index whose alias appears at the beginning of this entry.

**Control Volume Pointer Entry (CVPE)**

A control volume pointer entry can appear only in volume indexes. Two forms are possible: the old form, created prior to Release 17, and the new form, created since Release 17. Both forms are shown here.

**Old CVOL Pointer Entry**

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME	Name field; contains a high-level name that appears in the volume index of the control volume identified at offset 12 of this entry.
8 (8)	3	ETTR X'000000'	Zero field.
11 (B)	... 1	ETYPE X'03'	Type; indicates that this is either an old CVOL pointer entry (CVPE), or an index control entry (ICE). An ICE always appears as the first record of an index level; a CVOL pointer entry always appears in the volume index. This is also the number of halfwords that follow in the remainder of the entry.
12 (C)	6	EVOLIDO	Serial number of the control volume whose volume index contains an entry for the name found at the beginning of this entry.

**New CVOL Pointer Entry**

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME	Name; contains a high-level name that appears in the volume index of the control volume identified at offset 12 of this entry.
8 (8)	3	ETTR X'000000'	Zero field.
11 (B)	... 1	ETYPE X'05'	Type; indicates that this is a new CVOL pointer (CVPE) or the volume index control entry. The volume index control entry always appears as the first entry in the first block of SYSCTLG; a CVOL pointer never appears as the first entry of the first block. Also indicates that five halfwords follow in the remainder of the entry.
12 (C)	4	EDEV TYP	Control volume device type; contains the binary device code of the control volume whose volume index contains an entry for the name found at the beginning of this entry.
16 (10)	6	EVOLID	Serial number of the control volume whose volume index contains an entry for the name found at the beginning of this entry.



**Data Set Pointer Entry (DSPE)**

A data set pointer entry can appear in any index level. It contains the simple name of a data set and from one to five 12-byte fields that each identify a volume on which the named data set resides.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME	Name; contains the simple name of the data set whose volumes are identified at offset 12 of this entry.
8 (8)	3	EDSCBTTR	Address; contains either binary zero or, when the data set resides on only one volume, the relative track address (TTR) of the data set control block (DSCB) for this data set in the volume table of contents (VTOC).
11 (B)	. . . 1	ETYPE X'07' X'0D' X'13' X'19' X'1F'	Type; indicates that this is a data set pointer entry (DSPE). Also indicates the number of halfwords that follow in the remainder of this entry.
12 (C)	2	EVOLCNT	Volume count; contains the binary count of the number of volumes identified beginning at offset 14.
14 (E)	. . 12 to 60	EDATA	Volume entries; contains from one to five 12-byte entries, each of which identifies one volume on which the data set resides. Catalog management neither uses nor checks the contents of this field.

**Generation Index Pointer Entry (GIPE)**

A generation index pointer entry can appear in any index except a generation index. It corresponds to the simple name used in the relative name for a GDG data set.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME	Name; contains the name of the generation index to which this entry points.
8 (8)	3	ETTR	Address; contains the relative track address of the first block of the generation index named in this entry, in the form TTR.
11 (B)	. . . 1	ETYPE X'02'	Type; indicates that this is a generation index pointer entry (GIPE). Also indicates that two half-words follow in the remainder of this entry.
12 (C)	1	EGFLAGS	Flags; contains the options specified by the creator of the generation data group:  .... ..1. DELETED option. .... ..1. EMPTY option.
13 (D)	. 1	EGMAXSIZ	Maximum count; contains a binary number specifying the maximum number of generations allowed in the generation index at one time.
14 (E)	. . 2	EGCURSIZ	Current generation count; contains the binary number of generations currently cataloged in the index.

### Index Control Entry (ICE)

The index control entry is the first entry in all indexes except the volume index.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	INAME X'00...01'	Name; low value of binary one ensures that this is the first entry in the index.
8 (8)	3	ILSTBLK	Last block address; contains the relative track address of the last block assigned to the index, in the form TTR.
11 (B)	... 1	ITYPE X'03'	Type; indicates that this is either an ICE or an old CVOL pointer. An ICE always appears as the first entry of an index; an old CVOL pointer always appears in the volume index. Also indicates the number of halfwords that follow in the remainder of the entry.
12 (C)	3	IFSTBLK	First block address; contains the relative address of the block in which this entry appears, in the form TTR.
15 (F)	... 1	ILIASCNT	Number of aliases; contains a binary count of aliases assigned to the index. This count is always zero for indexes that are not high-level. An index cannot be deleted if this count is non-zero.
16 (10)	2		Reserved.

### Index Link Entry (ILE)

An index link entry is always the last entry in any index block. It is used to link blocks of one index into a chain.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME X'FF...FF'	Name; high value (all bits on) ensures that this is the last entry in the index.
8 (8)	3	ETTR	Link address; contains the relative track address of the next block of the same index, if there is one, in the form TTR. When this is the last (or only) block, this field contains binary zero.
11 (B)	... 1	ETYPE X'00'	Type; indicates that this is either an ILE or an IPE. The name field of an ILE always contains X'FFFFFFFFFFFFFFFF'; the name field of an IPE never does. Also indicates that there are no more halfwords in the entry.

### Index Pointer Entry (IPE)

The index pointer entry can appear in any index except a generation index. It points to a lower index.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME	Name; contains the name of the index to which this entry points.
8 (8)	3	ETTR	Index address; contains the relative track address of the first block of the index named in this entry, in the form TTR.
11 (B)	... 1	ETYPE X'00'	Type; indicates that this is either an IPE or an ILE. The name field of an ILE always contains X'FFFFFFFFFFFFFFFF'; the name field of an IPE never does. Also indicates that there are no more bytes in the entry.

**Volume Control Block (VCB)**

A volume list can be recorded in one or more volume control blocks. Each volume control block is one block of the SYSCTLG data set, and can identify up to 20 volumes on which one data set is recorded.

**Note:** This block is different from other blocks of SYSCTLG. The first halfword does not contain the number of bytes used in the block as do other SYSCTLG blocks. The field VCBVOLCT, shown below, is the first halfword of the VCB block.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	2	VCBVOLCT	Number of volumes; contains the number of volumes identified in this and subsequent volume control blocks. This number is reduced by 20 for each subsequent volume control block. For example, if a data set resides on 61 volumes, it uses four volume control blocks. This field of each block contains 61, 41, 21 and 1, respectively.
2 (2)	.. 12 to 240	VCBVOLS	Volume identifications; contains from 1 to 20 12-byte entries, each of which identifies one of the volumes on which the data set resides. Catalog management neither uses nor inspects the content of these entries. Each 12-byte entry contains a 4-byte device code, a 6-byte volume serial number, and a 2-byte data set sequence number.
242 (F2)	.. 10	X'00...00'	Zero field.
252 (FC)	3		Chain address, contains the relative track address of the next volume control block, if there is one, in the form TTR. If this is the last (or only) block of the volume control block, this field contains binary zero.
255	1	X'00'	Zero field.

**Volume Control Block Pointer Entry (VCBPE)**

A volume control block pointer entry can appear in any index. It is used when a data set resides on more than five volumes.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	ENAME	Name; contains the simple name of the data set whose volumes are identified in the volume control block that is pointed to by this entry.
8 (8)	3	ETTR	Address; contains the relative track address of the volume control block identifying the volumes containing the data set named in this entry, in the form TTR.
11 (B)	... 1	ETYPE X'01'	Type; indicates that this is a volume control block pointer entry. Also indicates that one halfword follows in the remainder of this entry.
12 (C)	2	X'0000'	Zero field.

**Volume Index Control Entry (VICE)**

The volume index control entry is always the first entry in the first block of data set SYSCTLG.

It is the control record for the entire data set, and acts as an ICE for the volume index.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8	VNAME X'00...01'	Name; always contains a binary one to ensure that this is the first entry of the volume index.
8 (8)	3	VLSTBLK	Last block address; contains the relative track address of the last block of the volume index, in the form TTR.
11 (B)	. . . 1	VTYPE X'05'	Type; indicates that this is the volume index control entry or a new CVOL pointer entry. The volume index control entry is always the first entry of the first block of SYSCTLG; a CVOL pointer is never the first entry. Also indicates that five halfwords follow in the remainder of the entry.
12 (C)	3	VCLSTBLK	Last block of the catalog; contains the relative track address of the last block in SYSCTLG, in the form TTR.
14 (E)	. . 1	VHIREC	Record number of TTR in VCLSTBLK. Note that this field is the last byte of VCLSTBLK (offset 12).
15 (F)	. . . 1	X'00'	Zero field.
16 (10)	3	VFHOLE	First available block; contains the relative track address of the first unused block in SYSCTLG, in the form TTR.
19 (13)	. . . 1	X'00'	Zero field.
20 (14)	2		Reserved.

**Environment Record (EREC DSECT)**

The environment record is written by module IGG0CLC7 under certain error conditions. This record is useful in diagnosing problems with the catalog-management routines. Reading the environment record is described in the chapter "Diagnostic Aids" on page 52.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	8		Reserved.
8 (8)	8	ERTIME	Time stamp, as produced by the TIME macro instruction.
16 (10)	4	ERCAMLST	First four bytes of the caller's parameter list produced by the CAMLST macro instruction (see description of CAMLSTD on page 53).
20 (14)	1	ERMODMAP	Field MODMAP1 from WORKAREA, described on page 57.
21 (15)	. 1	ERFLAG1	Field FLAG1 from WORKAREA, described on page 55.
22 (16)	. . 1	ERFLAG2	Field FLAG2 from WORKAREA, described on page 56.
23 (17)	. . . 1	ERFLAG3	Field FLAG3 from WORKAREA, described on page 56.
24 (18)	2	ERERRCOD	Fields ERRCATSV and ERRLOCSV from WORKAREA, described on page 55.
26 (1A)	. . 14	ERNAMTTR	Level name, TTR, type, and volcnt; the first fourteen bytes of a general entry.
40 (28)	60	ERREGSV	Contents of general registers 0 through 14 at the time the environment record is written (register 15 is destroyed by module IGG0CLC7).

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
100 (64)	28	ERWA1	Contents of WORKAREA from offset 12 bytes (label TTR) through offset 39 bytes.
128 (80)	18	ERINPUT	First entry in INPUT, a field described on page 58.
146 (92)	. . 18	EROUTPUT	First entry in OUTPUT, a field described on page 59.
164 (A4)	8	EROPTNCC	Field OPTNCCW from WORKAREA, described on page 59 .
176 (B4)	40	ERIOB	Field IOB from WORKAREA, described on page 59 .
212 (D4)	44	ERNAME	Fully qualified name provided by the caller.

## CAMLSTD DSECT

CAMLSTD describes the parameter list provided by the caller of catalog management. It maps the result of the CAMLST macro instruction.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	1	CAMOPTN1	First option byte. 1... .... Catalog is not on SYSRES. ..1. .... CAT or CATBX request. ...1 .... RECAT request. .... 1... UNCAT or UCATDX request. .... ..1. Locate-by-block request.
1 (1)	. 1	CAMOPTN2	Second option byte. .1.. .... BLDX or CATBX request. ..1. .... BLDG request. ...1 .... BLDA request. .... 1... LNKX request. .... ..1. DLTX or UCATDX request. .... ..1. DSCB TTR has been specified. .... ...1 DLT A request.
2 (2)	. . 2	CAMOPTN3	Third option byte. 1... .... DRPX request. .1.. .... Scratch GDG data sets. .... 1... Empty generation index when maximum generation count is reached.
3 (3)	. . . 1	CAMGEN	Maximum generation count.
4 (4)	4	CAMPTR1	Address of the name field in caller's area. For locate-by-block, the name field contains a relative track address instead of a name.
8 (8)	4	CAMCVOLP	Address of CVOL identification field (a 10-byte field containing the device type and volume serial number of the CVOL).
12 (C)	4	CAMPTR3	Address of caller's third parameter. Meaning depends on the function:  Locate            Caller's 265-byte work area BLDA             8-byte name field LNKX             10-byte volume identification  CAT, CATBX or RECAT        Volume list Other             Not used
16 (10)	4	CAMDSCBP	Address of three-byte field containing the relative track address (TTR) for the Format 1 DSCB for the data set named through CAMPTR1.

## RPSD DSECT

RPSD describes the CCW chain used for rotational positioning sensing (RPS) support.

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	16	RPSCCW	Two double-words: RPSSS and RPSTIC.
0 (0)	8	RPSSS	Set sector CCW.
8 (8)	8	RPSTIC	TICs to normal channel program.
16 (10)	16	RPSINPUT	Four words: RPSCNV, RPSDDKR, RPSR1, and RPSPTR.
16 (10)	4	RPSCNV	Address of supervisor routine to convert sector value.
20 (14)	4	RPSDDKR	Block size (DD, 256 bytes), key length (K, 8 bytes), and record number.
24 (18)	4	RPSR1	Address of location of this DSECT during use.
28 (1C)	4	RPSPTR	Type and address: the first byte contains the device type code, and the last three bytes contain the sector value.
32 (20)	40	RPSAVE	10-word register save area.

## WORKAREA DSECT

WORKAREA serves all catalog modules. WORKAREA serves as intermediate storage, communications area, and buffers. BLDLAREA is a portion of WORKAREA that serves the resident BLDL routines. For a locate function, BLDLAREA is separate from WORKAREA.

Many of the fields in the area overlay other fields, and sections of an area can have more than one label. Figure 9 shows where these overlays occur, by label.

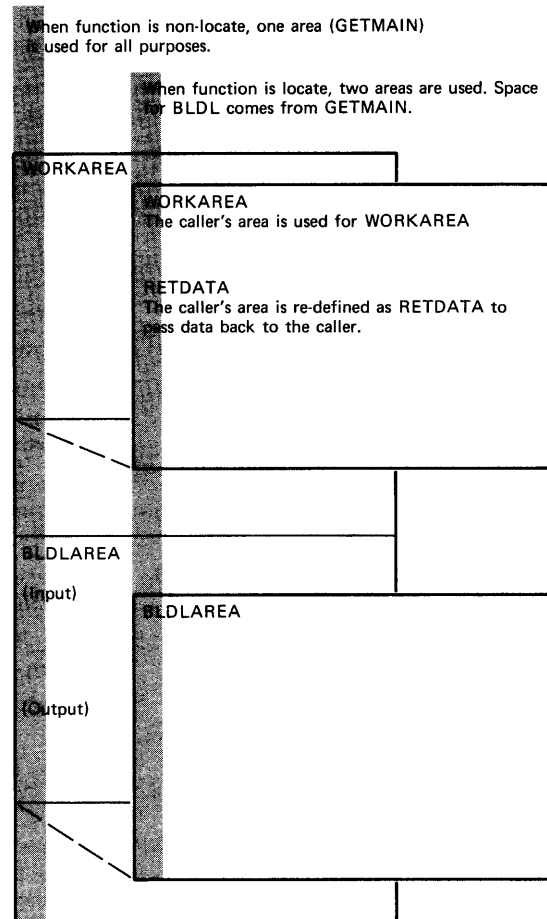


Figure 9. Data Area Hierarchy

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
0 (0)	4	BLDLIST LASTLEV	List parameter for BLDL or, when appropriate, the name of the last valid index level.
4 (4)	8	NAME ALIASNAM	Name in the entry that is being operated on, or the alias.
5 (5)	. 4	GENNO	Generation number portion of an absolute GDG name.
12 (C)	3	TTR	Relative track address in the current entry, in the form TTR.
15 (F)	. . . 1	TYPE	Type of entry; also the binary number of halfwords following in the remainder of the entry. TYPE is interpreted:
		X'00'	Either an index pointer entry (IPE) or an index link entry (ILE). The name field of an ILE always contains X'FFFFFFFFFFFFFFFF', the name field of an IPE never does.
		X'01'	Volume control block pointer entry (VCBPE).
		X'02'	Generation index pointer entry (GIPE).
		X'03'	Index control entry (ICE) or old CVOL pointer entry (CVPE). An ICE always appears as the first entry of an index, a CVPE always appears in the volume index.
		X'04'	Alias entry (AE).
		X'05'	Volume index control entry (VICE) or new CVOL pointer entry (CVPE). The VICE always appears as the first entry of the first block of the catalog; a CVPE always appears later in the volume index.
		X'07'	Data set pointer entry (DSPE with one volume identification).
		X'0D'	DSPE with two volumes.
		X'13'	DSPE with three volumes.
		X'19'	DSPE with four volumes.
		X'1F'	DSPE with five volumes.
16 (10)	8	TRUE	The true name to which the alias in offset 4 relates.
16 (10)	2	VOLCNT	Number of volumes identified in DATA when the current entry is a data set pointer entry (DSPE).
16 (10)	62	DATA	Volume identifications for DSPE.
88 (58)	1	ERRCATSV	Error code generated for non-locate function.
89 (59)	. 1	ERRLOCSV	Error code generated for locate function.
90 (5A)	. . 1	FLAG1	Switches declaring requested function.
		.1.. ....	The index control entry (ICE) must be read.
		..1. ....	SYSCTLG has no more room during CATBX or BLDX function.
		...1 ....	The DCB/DEB was freed by SVC 28 processing.
		.... 1...	CATBX request.
		.... .1..	UCATDX request.
		.... ..1.	Locate request.
		.... ....1	RECAT request.

*Continued*

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
91 (5B)	. . . 1	FLAG2	Switches used to specify flow of control.
		..1. ....	RPS device.
		.... 1...	'(' has been found in the data set name.
		.... .1..	Last entry found was a CVOL pointer entry (CVPE).
		.... ..1.	Generation index pointer entry (CIPE) has been found.
92 (5C)	28	SAVEAREA	Save area for temporarily storing the contents of general purpose registers.
120 (78)	8	NEXTKEY NEXTCNT	The key or count of the next block beyond the one read.
128 (80)	10	ICE	Index control entry; see the description on page 50 for individual fields. Only bytes 8 through 15 are saved here.
136 (88)	3	VICESAVE	Save area for volume index control entry information.
139 (8B)	. . . 9	VICE	Volume index control entry; see the description on page 52 for individual fields. Only bytes 11 through 18 are saved here.
148 (94)	1	FLAG3	Switches to invoke functions of IGG0CLC5.
		1... ....	Absolute GDG name found.
		.1.. ....	Free index blocks.
		..1. ....	Read a block for updating.
		...1 ....	Process EMPTY option of generation data group (GDG).
		.... 1...	Write the last block of a GDG chain when the GDG is full and a new one is being added.
		.... .1..	Build volume control blocks (VCBs).
		.... ..1.	Free VCBs.
		.... ...1	Write a block.
149 (95)	. 1	FLAG4	Switches to control the flow of control in IGG0CLC6.
		1... ....	New entry has been inserted into block now in the work area. Updating is in process.
		.1.. ....	The updated block has been written into SYSCTLG. Updating is complete.
		...1 ....	The block following the block pointed to by field WRITETTR is free.
		.... ..1.	The first write has occurred.
		.... ...1	The block following the block pointed to by field LINKTTR is free.
150 (96)	. . 2	NAMLEN	Length of the full name given by caller minus 1.
152 (98)	4	NAMDELMP	Address of last delimiter in given name.
156 (9C)	4	NAMLSTP	Pointer to last displacement of given name in the name table.

*Continued*



<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
162 (A2)	.. 1	MODMAP1	Trace of modules that have been entered. The appropriate bit is set to 1 as each module is entered. There is no bit for module IGC0002F, because this module is always entered before any other.
		1... ....	IGG0CLC1
		.1.. ....	IGG0CLC2
		..1. ....	IGG0CLC3
		...1 ....	IGG0CLC4
		.... 1...	IGG0CLC5
		.... .1..	IGG0CLC6
		.... ..1.	IGG0CLC7
164 (A4)	4	EPBLDL	Address of the entry point of the supervisor routine BLDL, IECPBLDL (copied from field CVTPBLDL of the CVT).
168 (A8)	4	BLDLISTP	Address of the list to be completed by BLDL (address of field BLDLIST, offset 0 of this DSECT).
172 (AC)	4	DCBADDR	Address of the data control block (DCB) for the control volume.
176 (B0)	4	DEBADDR	Address of the data extent block (DEB) for the control volume.
180 (B4)	4	FOUNDENT	Address of an entry in an input/output buffer.
184 (B8)	4	EPTORLTV	Address of the entry point IECPRLTV, a supervisor routine that converts absolute track addresses to relative track addresses (copied from field CVTPRLTV of the CVT).
188 (BC)	4	EPTOABSL	Address of the entry point IECPCNVT, a supervisor routine that converts relative track addresses to absolute track addresses (copied from field CVTPCNVT of the CVT).
192 (C0)	4	SVRBEXTP	Address of the extension to the SVRB.
196 (C4)	4	ADDING	Address of new entry, meaningful only when bit 0 of FLAG4 is '1'.
200 (C8)	16	XCTLPARM	Parameters for XCTL SVC.
216 (D8)	12	LNKENTRY	General form in index link entry (ILE). The first eight bytes contain X'FFFFFFFFFFFFFFFF'.
224 (E0)	4	LINKTTR	Last four bytes of LNKENTRY; contains the TTR for this ILE.
228 (E4)	4	WRITETTR	Save area for relative address of block to be written.
232 (E8)	4	SAVETTR2	Second save area for any relative track address.
236 (EC)	4	ICETTR	Relative track address of block that contains an index control entry (ICE).
240 (F0)	4	SAVETTR	Save area for any relative track address.
244 (F4)	4	READTTR	Save area for relative address of block to be read.

*Continued*

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
248 (F8)	2	NAMLF	Number of levels of the name that were found.
250 (FA)	. . 2	NAMLG	Number of levels in given name.
252 (FC)	4	DEVTYPE	Device-type portion of an identification.
256 (100)	1	THETA	Angular displacement value (theta) for rotational positioning support (RPS).
257 (101)	. 1	INDEXLEN	Length of all levels given except the last. Used with SCRATCH macro instruction.
258 (102)	. . 1	ERRSV2H	Exceptional return code from module IGC0002H.
259 (103)	. . . 6	VOLSN	Serial-number portion of a volume identification.
16 (10)	44	DSNAME	Data set name to be scratched (when processing GDG data sets).
60 (3C)	12	SCRPARM	Parameter list for SCRATCH macro instruction.
72 (48)	4	SCRVOLS	Volume list for SCRATCH macro instruction.
32 (20)	44	NAMTABLE	Name table containing the length and displacement of each component of the given name.
76 (4C)	. 1	NAMDELIM	Last delimiter in the given name, either 'b' or '('.
128 (80)	8	RELNUMBR	Work area for Convert-to-Binary instruction used with relative GDG processing.
136 (88)	8	PKDNUMBR	Work area for Pack instruction used with relative GDG processing.
0 (0)	256	RETDATA	Volume list returned to caller.
252 (FC)	4	REDSGBT	Relative track address of the DSCB in the VTOC for a single-volume data set, as returned to the caller.
259 (103)	. . . 6	RETCVOL	Serial number for the control volume containing the returned volume list.
268 (10C)	4	BALREGS	Save area for register used in BAL instruction.
272 (110)	400	BLDLAREA	Work area for use by BLDL routine; for a locate function, WORKAREA is in two parts, and BLDLAREA is the second part.
528 (210)	120	BLDLCNT	Parameters for BLDL routine.
648 (288)	4	BASESAVE	Save area for the register that would otherwise be destroyed by BLDL.
536 (218)	44	RESALIAS	Work area used when resolving an alias name.
272 (110)	256	INPUT	Input buffer for channel program.
272 (110)	256	TRTABLE	Translate table used with TR instruction to analyze the given name.
528 (210)	8	SIDE1	Search-ID-Equal CCW.

*Continued*

<i>Offset</i>	<i>Bytes and Alignment</i>	<i>Name and Content</i>	<i>Field Description and Meaning</i>
536 (218)	8	TIC1	TIC CCW.
544 (220)	8	OPTNCCW	CCW that is changed to do the required input/output function.
552 (228)	8	RC	Read-Count CCW.
560 (230)	8	SKE	Search-Key-Equal CCW.
568 (238)	8	TIC2	Transfer-In-Channel CCW.
576 (240)	8	NOP	NOP CCW.
584 (248)	4	ECB	Event control block for channel programs.
588 (24C)	40	IOB	Input/output block for channel programs.
360 (168)	8	RKD	Read-Key-Data CCW.
638 (27C)	8	RD	Read-Data CCW.
644 (284)	8	WKD	Write-Key-Data CCW.
652 (28C)	264	OUTPUT	Output buffer for channel programs.

## Diagnostic Aids

This chapter provides several aids that can be useful when diagnosing difficulties with the catalog management modules. These aids tell you:

- What to look for in a main storage dump.
- How to dump the SYSCTLG data set.
- A description of an *environment record* that is written when an error is detected.
- How to determine what modules are involved in a particular case and what the flow of logic is between them.

### Reading Dumps

Two kinds of dumps can be used while diagnosing trouble with catalog-management routines: main storage dumps and SYSCTLG data set dumps. This section points out significant diagnostic clues to look for. It does not explain the full meaning of dumps; for that information, see *Guide to Reading OS System Dumps*, Order Number GC28-6670.

### Main Storage Dump

Cases when a main storage dump is helpful are rare. Unless the error that caused the ABEND was actually in one of the catalog-management routines, the transient areas will probably have been used by another module of the operating system.

If, however, an ABEND dump was produced because of an error in one of the catalog modules, then look at the content of the general registers at the time of the ABEND. Most significant are:

- Register 4     Base register for all catalog-management routines. This register points to the module that failed.
- Register 6     Pointer to WORKAREA (described on page 54). The field MODMAP1 shows which modules have been entered; compare this to the expected path for the requested function. The section “Module Selection Charts,” on page 62, shows the path for each function.
- Register 8     Pointer to caller’s CAMLST, described on page 53. Check the parameters in CAMLST carefully to be sure that the requests are valid.

Have the following items available before requesting additional assistance:

- Source or input listing for the use of catalog-management routines (be sure MSGLEVEL=(1,1)).
- Main storage dump produced by using a SYSABEND DD statement.
- Dump of SYSCTLG data set.

### SYSCTLG Dump

There are several ways to dump a data set; this discussion assumes IMASPZAP, a service-aid program that operates under the operating system. IMASPZAP is described in *OS Service Aids*, Order Number GC28-6719.

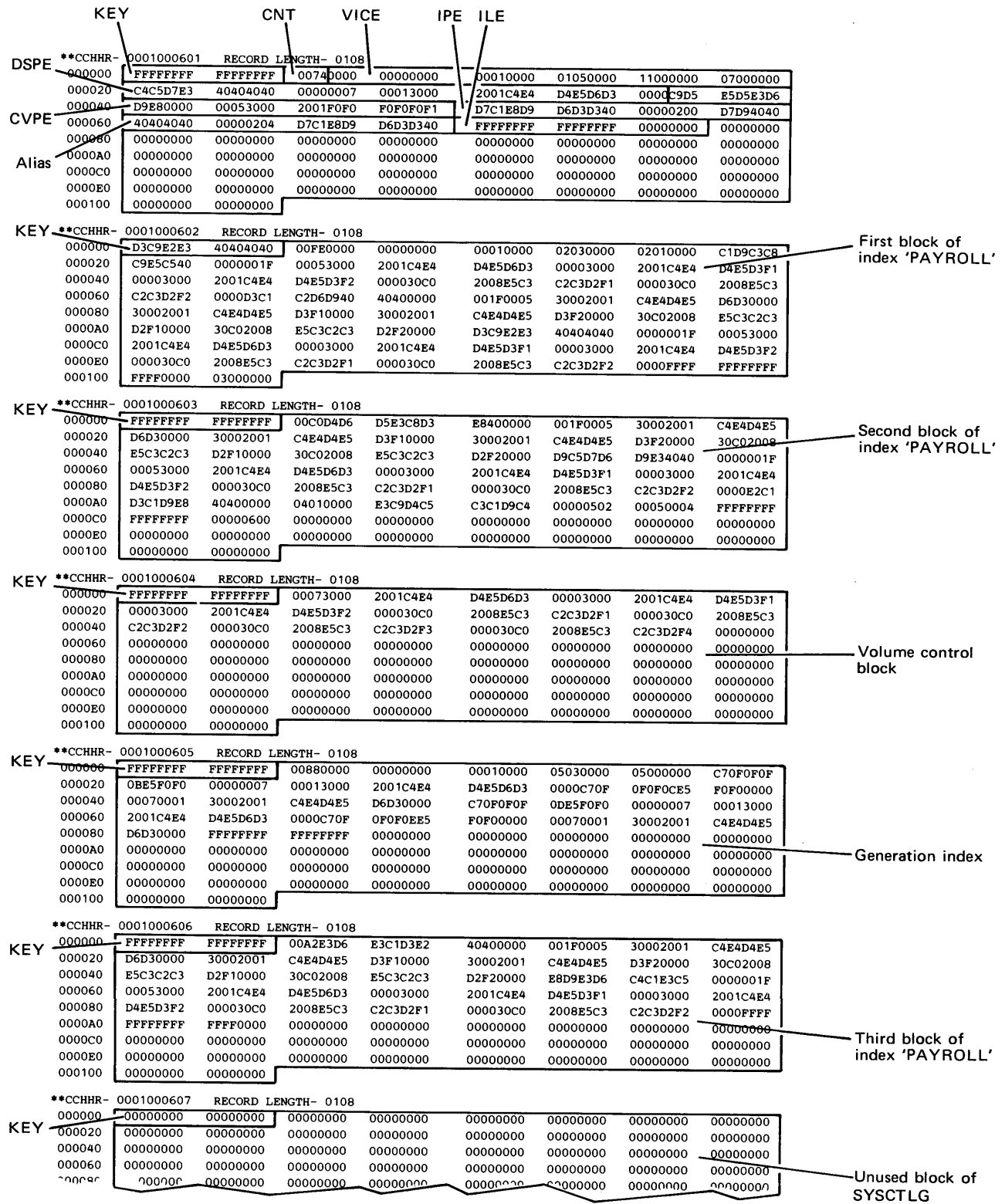
To dump the catalog with IMASPZAP, use the following JCL, where the SYSLIB DD card points to the CVOL to be dumped:

```
//DUMPSTEP EXEC PGM=IMASPZAP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=SYSCTLG,UNIT=xxxx,
// VOL=SER=xxxxxx,DISP=OLD,
// DCB=(KEYLEN=8)
//SYSIN DD *
  ABSDUMP ALL
/*
```

This JCL is used to dump the entire catalog; you can dump a portion by specifying beginning and ending track addresses. The DCB parameter KEYLEN in the SYSLIB DD statement formats the key as well as the data for each block. The key appears as the first two words of the first line of each block. The data for the block begins in the third word.

**Example of a  
SYSCTLG Dump**

Figure 10 shows an actual dump of the catalog. Entries in the volume index are outlined, and other blocks of the catalog are identified.



**Figure 10. Example of a SYSCTLG Dump**

The data portion of each block begins with a 16-bit binary number that tells how many bytes of the block are used (including the two bytes of this number). Immediately thereafter begin the entries of the catalog. These entries are described in detail in the chapter "Data Areas," page 47.

The first entry of the first block is always the volume index control entry (VICE). The type of each entry can be determined from the byte at offset 11 of the entry; the type codes are described under the field TYPE on page 55.

## Environment Record

Some error conditions cause an environment record to be written to the last block of SYSCTLG, whenever possible. The environment record is written when a non-locate function is requested and the exceptional return code is 8 or 28. Here's how you can dump the catalog and examine this record to see what happened:

1. Reproduce the failure, but this time reserve the data set SYSCTLG for your exclusive use, so that no other task can destroy the environment record before you can dump it. Do this by adding or modifying your JCL statements to include a DD statement for SYSCTLG with DISP=OLD.
2. Add a step to your job to dump SYSCTLG. Follow the instructions under "Dumping SYSCTLG," page 60.
3. Look at the VICE, which begins at offset two of the first physical block of the catalog. (Remember to allow for the key.) Field VCLSTBLK (offset 12 bytes in the VICE) contains the TTR for the last block in SYSCTLG. This block contains the environment record.
4. Compute the absolute track address by using the cylinder-head numbers supplied for the first block and the TTR. TT is the relative track from the first block; R is the record number for that track.
5. The fields of the environment record are described in "Environment Record," on page 52. The description for each field relates this information to other data areas.

The field ERMODMAP contains seven bits that show which modules have been entered. IGC0002F is always entered; there is no bit switch for that module.

As an example, if ERMODMAP equals X'66', then modules IGC0002F, IGG0CLC2, IGG0CLC3, IGG0CLC6, and IGG0CLC7 were entered during the request that caused the environment record to be written. This is the sequence of modules that normally occurs with a request for BLDX.

**Note:** The environment record is not written for any error associated with a "catalog-full" condition.

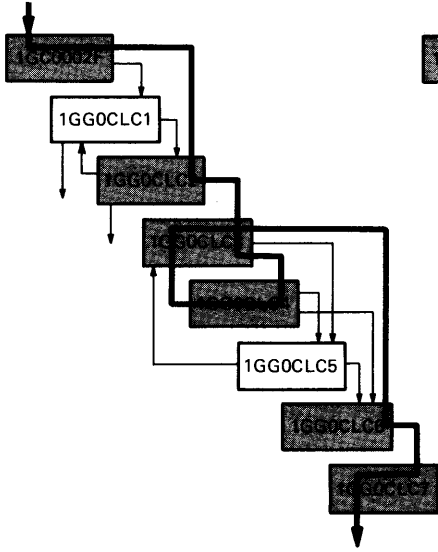
## Module Selection Charts

Figure 11 can help you determine which modules of the catalog management routines are involved in any given situation. The figure consists of several charts, which are modified reductions of Figure 6, found on page 21. Each chart shows the path through the catalog-management routines for the functions noted on that chart. The blocks that are used have bold outlines; the specific path is shown by a wide, shaded arrow. Always enter module IGG0002F (upper-left), and then move down and to the right.

Page number references for flowcharts and module discussions are:

<i>Module Name</i>	<i>Page Number</i>
IGC0002F	26
IGG0CLC1	28
IGG0CLC2	30
IGG0CLC3	32
IGG0CLC4	34
IGG0CLC5	36
IGG0CLC6	38
IGG0CLC7	40
IGC0002H	42
IGG0CLF2	44

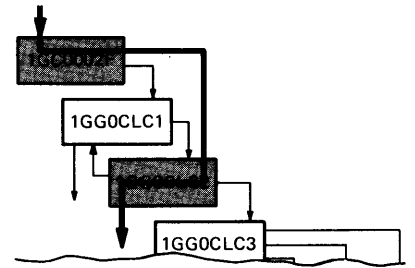
CATBX



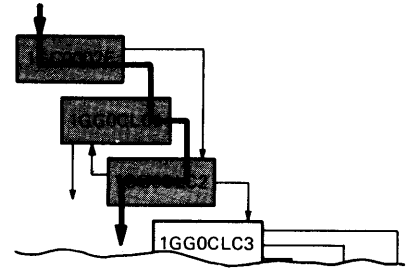
CAT, UNCAT, RECAT



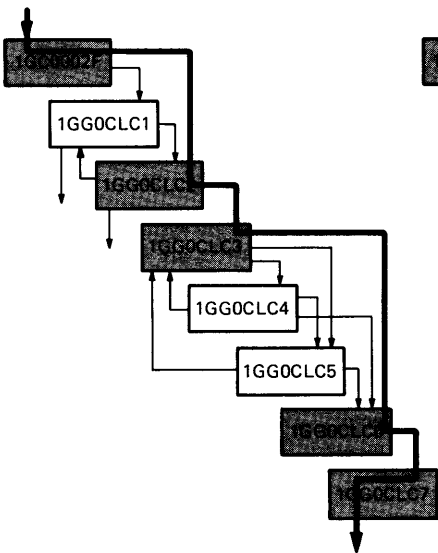
Locate Simple or Qualified Name



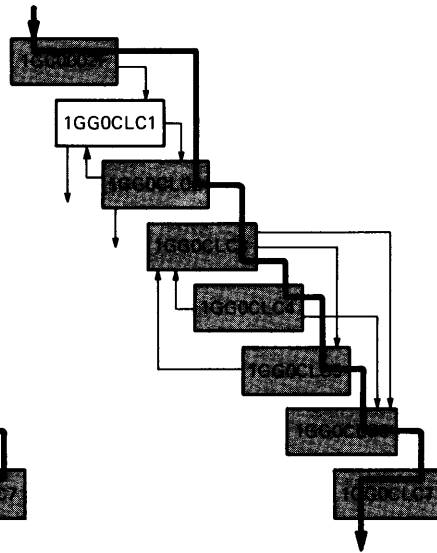
Locate Qualified Name with an Alias



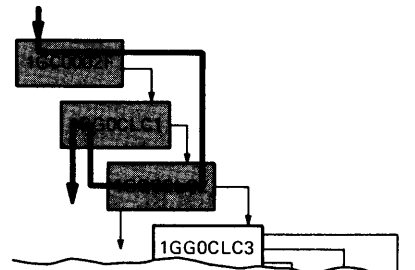
DLTX, UCATDX, LINKX, DRPX, BLDA, DLTA



Catalog Function with VCB Processing Required, or GDG Empty Option Required



Locate Relative GDG Name



Locate Relative GDG Name with an Alias

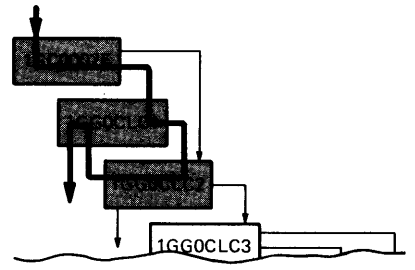


Figure 11. Correlating Functions to Modules of Catalog Management

## Glossary

This glossary contains definitions of words and acronyms that are used in this publication. Other data processing definitions can be found in *A Data Processing Glossary*, Order Number GC20-1699.

**alias:** An alternative name for a data set. Only the high-level name of a qualified data set name may have an alias.

**CAMLST:** An Assembler language macro instruction that produces the parameter list for catalog management.

**catalog:** (noun) The collection of all data set indexes maintained by data management. (verb) To include the volume identification of a data set in the catalog.

**CATALOG:** An Assembler language macro instruction used to invoke the catalog-management routines.

**cataloged data set:** A data set that is represented in an index or hierarchy of indexes that provides the means for locating the data set.

**communication vector table (CVT):** An operating system control block that provides the address of information in the nucleus to non-resident routines.

**connected:** The property of a control volume that allows it to be found by searching another control volume.

**control volume (CVOL):** A volume that contains one or more of the indexes of the catalog.

**data control block (DCB):** An operating system control block that describes the current use of the data set.

**data extent block (DEB):** A control block that describes the physical attributes of the data set.

**data set:** The major unit of data storage and retrieval in the operating system.

**data set control block (DSCB):** A label for a data set on a direct storage volume.

**data set name:** An identifier that unambiguously names a data set.

**data set pointer entry (DSPE):** A catalog entry that identifies the volume on which a named data set resides.

**dequeue:** To remove a request for a resource from a list of requests.

**enqueue:** To build a list of requests for a named resource.

**entry:** A logical record of the catalog, data set SYSCTLG.

**environment record:** A 265-byte record that is written when the catalog-management routines discover an error. This record, which contains significant data that is present at the time of the error, is written to the last block of data set SYSCTLG for later analysis.

**generation:** One member of a generation data group.

**generation data group (GDG):** A collection of historically related data sets.

**generation index:** An index of the catalog that identifies the generations of a generation data group.

**generation index pointer entry (GIPE):** A catalog entry that identifies a generation index.

**high-level name:** The first component of a qualified name. This name is found in a volume index of the catalog.

**index:** A table in the catalog structure that is used to locate data sets.

**INDEX:** An Assembler language macro instruction used to invoke the catalog management routines.

**index control entry (ICE):** The first entry of each index of the catalog. This entry contains all control information about the index.

**index link entry (ILE):** The last entry of each block of the catalog, used to link blocks of one index together in a chain.

**index pointer entry (IPE):** A catalog entry that attaches a lower-level index to the index in which it is found.

**level:** A conceptual relationship between indexes of the catalog. The index corresponding to the simple name of a data set is said to be the lowest level; the first component of a qualifier name is said to correspond to the highest-level index.

**locate:** Pertaining to functions that do not change the status of the catalog; that is, read-only operations are performed on the catalog.

**LOCATE:** An Assembler language macro instruction used to invoke the catalog management routines.

**must-complete:** A facility of the operating system that can ensure integrity whenever system resources are manipulated to modify information that is critical to the continued operation of the system.

**non-locate:** Pertaining to functions that change the status of the catalog; that is, write operations are performed on the catalog.

**partitioned data set directory:** The portion of a partitioned data set that provides a means of locating any of the members of the data set.

**qualified name:** A data set name consisting of a string of names separated by periods; for example, "TREE.FRUIT.APPLE" is a qualified name.

**qualifier:** Each component names in a qualified name other than the rightmost name. For example, "TREE" and "FRUIT" are qualifiers in "TREE.FRUIT.APPLE".

**relative track address (TTR):** A direct-access device address, expressed as a displacement in a data set. This address has the form TTR, where TT represent two hexadecimal digits specifying the track relative to the beginning of the data set, and R is one hexadecimal digit specifying the record on that track.

**resource:** Any facility of the computing system or operating system required by a job or task, including main storage, input/output devices, the central processing unit, data sets, and control processing systems. Three unique resources are used by catalog-management routines: the volume index, high-level name, and the volume index control entry (VICE).

**ripple:** Displacing data from one block of a chain to the next, due to modification of data in a preceding block.



**simple name:** The rightmost component of qualified name. For example, "APPLE" is the simple name in "TREE.FRUIT.APPLE". The simple name corresponds to the lowest index level for the data set name.

**supervisor request block (SVRB):** An operating system control block containing program status information and general register contents.

**SYSCTLG:** The data set name of the catalog.

**system residence volume:** The volume on which the nucleus of the operating system and the highest level of the catalog are located.

**task control block (TCB):** An operating system control block that contains information and pointers associated with the task in progress.

**transient area:** An area of main storage contained within the operating system nucleus where supervisor routines may be temporarily loaded for execution.

**true name:** The high-level qualifier to which an alias is related.

**uncatalog:** To remove the volume identification of a data set from the catalog.

**volume control block (VCB):** A block of the catalog that identifies as many as 20 volumes containing one data set.

**volume control block pointer entry (VCBPE):** A catalog entry that identifies a VCB for a named data set.

**volume index:** The highest level of index in the catalog structure. Entries in the volume index point to all lower indexes and simple names.

**volume index control entry (VICE):** The first entry in the volume index. The VICE describes the volume index and controls space allocation in SYSCTLG.

**volume table of contents (VTOC):** A table associated with a direct access volume that describes each data set on that volume and identifies all available space on the volume.

# Index

## A

ABEND dump	60
absolute name	
complemented	11
generation data group	11
track addresses to relative, conversion	23
ADDING data field	57
adding entries	38
address conversion, track	23
aids for diagnosing	60
alias	
name	11
number of	11
resolution	28
Alias Entry (AE)	48
ALIASNAM data field	55
attribute, reentrant	23

## B

BALREGS data field	58
base register	24
BASESAVE data field	58
BLDA function	13
BLDENTRY routine in IGG0CLC4	35
BLDG function	13
BLDL macro instruction	
listed in program organization	23
in method of operation	14
BLDLAREA data field	58
BLDLCNT data field	58
BLDLIST data field	55
BLDLISTP data field	57
BLDX function	13
BLOCK function	13
blocks	
chaining	9
contiguous	9
data	9
how chained together	14
multiple	9
relation to index levels	9
use of key	9
BLVCBRTN routine in IGG0CLC5	36
building a new entry	17
bytes and alignment of data areas	47

## C

CALLBLDL routine	
in IGG0CLC1	29
in IGG0CLC2	30
relation to BLDL	23
CAMCVOLP data field	53
CAMDSCBP data field	53
CAMGEN data field	53
CAMLST macro instruction	13
CAMLSTD DSECT, data area	53
CAMOPTN1 data field	53
CAMOPTN2 data field	53
CAMOPTN3 data field	53
CAMPTR1 data field	53
CAMPTR3 data field	53
CAT function	13

## catalog

data set	
( <i>see also</i> SYSCTLG)	
dump of	60
introduction	9
entries	11
management	
design of	14
functions of	13
input to	14
macros and services used	13
output from	14
overview of	14
phases in	21
residence	10
on system residence	10
CATALOG macro instruction	13
CATBX	
function	13
method of operation	17
chaining	
blocks together	9
method of operation	14
Character Dependency	24
collating sequence	9
Communication Vector Table (CVT)	24
complementing absolute names	11
connected control volume	10
constructing	
DSPE or VCBPE	34
new entries	17
contiguous blocks	
in description of index levels	9
method of operation	14
control volume (CVOL)	
connected	10
introduction	10
opening	14
specifying	10
Control Volume Pointer Entry (CVPE)	48
conversion of track addresses	23
core storage ( <i>see</i> main storage)	
correlating functions to modules	62
CVOL ( <i>see</i> control volume)	
CVT DSECT macro instruction	24
CVTORLTV, reference to	23
CVTPBLDL, reference to	23
CVTPCNVT, reference to	23

## D

data	
areas	
descriptions	47
format of	47
hierarchy	54
block	9
control block (DCB)	
definition	64
modifications to	42
use with catalog management	14
extent block (DEB)	
modifications to	42
use with catalog management	14

set control block (DSCB)	
use with catalog management . . . . .	9
set name . . . . .	9
set pointer entry (DSPE)	
constructing . . . . .	34
entry description . . . . .	12
format . . . . .	49
DATA data field . . . . .	55
DCB ( <i>see</i> data control block)	
DCBADDR data field . . . . .	57
DEB ( <i>see</i> data extent block)	
DEBADDR data field . . . . .	57
definitions ( <i>see</i> Glossary)	
DELETE option of generation data group . . . . .	34
deleting entries	
method of operation . . . . .	17
module description . . . . .	38
dependency, character . . . . .	24
DEQ macro instruction . . . . .	23
dequeue, definition . . . . .	64
description of catalog entries . . . . .	12
design of catalog management . . . . .	14
DEVTYPE data field . . . . .	58
Diagnostic Aids . . . . .	60
directory, microfiche . . . . .	46
DLTA function . . . . .	13
DLTX function . . . . .	13
DRPX function . . . . .	13
DSCB ( <i>see</i> data set control block)	
DSNAME data field . . . . .	58
dump	
environment record . . . . .	62
main storage . . . . .	60
SYSCTLG . . . . .	60
<b>E</b>	
ECB data field . . . . .	59
EDATA data field . . . . .	49
EDEVTYPE data field . . . . .	48
EDSCBTTR data field . . . . .	49
EGCURSIZ data field . . . . .	49
EGFLAGS data field . . . . .	49
EGMAXSIZ data field . . . . .	49
EMPTY option of generation data group . . . . .	36
EMPTYRTN routine in IGG0CLC5 . . . . .	36
ENAME data field	
alias entry . . . . .	48
CVOL pointer entry . . . . .	48
data set pointer entry . . . . .	49
generation index pointer entry . . . . .	49
index link entry . . . . .	50
index pointer entry . . . . .	50
volume control block pointer entry . . . . .	51
ENQ macro instruction . . . . .	23
enqueueing resources . . . . .	24
entry, SYSCTLG	
alias . . . . .	48
constructing a new . . . . .	17
control volume pointer . . . . .	48
data set pointer . . . . .	49
deleting . . . . .	17
formats . . . . .	47
generation index pointer . . . . .	49
index control . . . . .	50
index pointer . . . . .	50
introduction . . . . .	12
overflow . . . . .	17
volume control block pointer . . . . .	51
environment record	
EREC DSECT data area . . . . .	52
use in diagnosing . . . . .	62
writing . . . . .	40
EPBLDL data field . . . . .	57
EPTOABSL data field . . . . .	57
EPTORLTV data field . . . . .	57
equipment requirements . . . . .	9
ERCAMLST data field . . . . .	52
EREC DSECT data area . . . . .	52
ERERRCOD data field . . . . .	52
ERFLAG1 data field . . . . .	52
ERFLAG2 data field . . . . .	52
ERFLAG3 data field . . . . .	52
ERINPUT data field . . . . .	53
ERIOB data field . . . . .	53
ERMODMAP data field . . . . .	52
ERNAME data field . . . . .	53
ERNAMTTR data field . . . . .	52
EROPTNCC data field . . . . .	53
EROUTPUT data field . . . . .	53
ERRCATSV data field . . . . .	55
ERREGSV data field . . . . .	52
ERRLOCSV data field . . . . .	55
error exception codes	
IGC0002F . . . . .	27
IGC0002H . . . . .	43
IGG0CLC1 . . . . .	29
IGG0CLC2 . . . . .	31
IGG0CLC3 . . . . .	33
IGG0CLC4 . . . . .	35
IGG0CLC5 . . . . .	37
IGG0CLC6 . . . . .	39
IGG0CLC7 . . . . .	41
IGG0CLF2 . . . . .	45
ERRSV2H data field . . . . .	58
ERTIME data field . . . . .	52
ERWA1 data field . . . . .	53
ETRUEN data field . . . . .	48
ETTR data field	
alias entry . . . . .	48
CVOL pointer entry . . . . .	48
generation index pointer entry . . . . .	49
index link entry . . . . .	50
index pointer entry . . . . .	50
volume control block pointer entry . . . . .	51
ETYPE data field	
alias entry . . . . .	48
CVOL pointer entry . . . . .	48
data set pointer entry . . . . .	49
generation index pointer entry . . . . .	49
index link entry . . . . .	50
index pointer entry . . . . .	50
volume control block pointer entry . . . . .	51
EVOLCNT data field . . . . .	49
EVOLID data field . . . . .	48
EVOLID0 . . . . .	48

example	
alias	10
CATBX	17
control volume	10
generation data group	12
JCL to dump SYSCTLG	60
path through catalog management	23
qualified name	10
simple name	9
SYSCTLG	9
SYSCTLG dump	61
volume index	9
EXCP macro instruction	24
EXIT SVC instruction	24
extending SYSCTLG	42
<b>F</b>	
field description and meaning of data areas	47
finding SYSCTLG	9
FLAG1 data field	55
FLAG2 data field	56
FLAG3 data field	56
FLAG4 data field	56
flowcharts	
IGC0002F	26
IGC0002H	42
IGG0CLC	28
IGG0CLC2	30
IGG0CLC3	32
IGG0CLC4	34
IGG0CLC5	36
IGG0CLC6	38
IGG0CLC7	40
IGG0CLF2	44
format of data area descriptions	47
format of SYSCTLG entries	47
formatting SYSCTLG extent	44
FOUNDEnt data field	57
FRBLKRTN routine in IGG0CLC5	36
FREEMAIN macro instruction	24
FRNDXRTN routine in IGG0CLC5	36
FRVCBRTN routine in IGG0CLC5	36
functions of catalog management	13
<b>G</b>	
GDG ( <i>see</i> generation data group)	
generation	
data group (GDG)	
absolute name	11
complementing names	11
example	12
generation index	11
generation number	11
introduction	11
options	34
processing relative number	28
relative number	11
version number	11
index	
definition	64
pointer entry (GIPE)	
format	49
introduction	12
number	11
GENNO data field	55
GET routine	
in IGG0CLC4	35
in IGG0CLC6	38
GETMAIN macro instruction	24
GETMAINB routine in IGG0CLC7	40
Glossary	64
<b>H</b>	
hierarchy of data areas	54
high-level name	
introduction	9
resource enqueueing	24
<b>I</b>	
IBM 2321 Data Cell	10
ICE	
data field	56
data area ( <i>see</i> index control entry)	
ICETTR data field	57
IECPBLDL, use in catalog management	23
IECPCLTV, use in catalog management	23
IEFUCBOB DSECT macro instruction	24
IEHPROGM, relation to catalog management	12
IFSTBLK data field	50
IGC0002F	
flowchart	26
module description	26
relation to overall	21
IGC0002H	
flowchart	42
module description	42
relation to overall	21
IGG0CLC1	
flowchart	28
module description	28
relation to overall	21
IGG0CLC2	
flowchart	30
module description	30
relation to overall	21
IGG0CLC3	
flowchart	32
module description	32
relation to overall	21
IGG0CLC4	
flowchart	34
module description	34
relation to overall	21
IGG0CLC5	
flowchart	36
module description	36
relation to overall	21
IGG0CLC6	
flowchart	38
module description	38
relation to overall	21
IGG0CLC7	
flowchart	40
module description	40
relation to overall	21
IGG0CLF2	
flowchart	44
module description	44
relation to overall	21
IGG0553A, use in catalog management	42
IGG0553E, use in catalog management	42
ILIASCNT data field	50
ILSTBLK data field	50
IMASPZAP, use with catalog management	60
INAME data field	50

INCR routine			
in IGG0CLC4	35		
in IGG0CLC5	36		
index control entry (ICE)			
format	50		
introduction	12		
updating			
method of operation	17		
module	40		
index levels, relation to physical blocks	9		
index link entry (ILE)			
format	50		
introduction	12		
index pointer entry (IPE)			
format	50		
introduction	12		
index			
generation	11		
introduction	9		
volume	9		
INDEX macro instruction	13		
INDEXLEN data field	58		
initializing a PDS directory	44		
initializing SYSCTLG	44		
input to catalog management	14		
INPUT data field	58		
IO routine			
in IGG0CLC3	33		
in IGG0CLC4	35		
in IGG0CLC5	37		
in IGG0CLC6	36		
in IGG0CLC7	41		
use of	24		
IOB data field	59		
ITYPE data field	50		
<b>J</b>			
Job Control Language (JCL)			
to dump SYSCTLG	60		
invoking catalog management	13		
job scheduler	13		
<b>K</b>			
key of SYSCTLG data blocks			
equal to data set name	17		
how used	9		
X'FFFFFFFFFFFFFFFF'	14		
X'0000000000000000'	17		
KEYICE routine in IGG0CLC3	33		
<b>L</b>			
LASTLEV data field	55		
length of name, limitations	9		
level, index	9		
limit			
aliases	11		
name lengths	9		
SYSCTLG residence devices	10		
LINKTTR data field	57		
LNKENTRY data field	57		
LNKX function	13		
locate			
list of functions	13		
method of operation	14		
LOCATE macro instruction	13		
<b>M</b>			
machine requirements	9		
macros for invoking catalog management	13		
Macros and Services Used	23		
main storage			
dump	60		
requirements	9		
memory ( <i>see</i> main storage)			
Method of Operation	14		
Microfiche Directory	46		
modifications to DCB/DEB	42		
module			
descriptions	24		
IGC0002F	26		
IGC0002H	42		
IGG0CLC1	28		
IGG0CLC2	30		
IGG0CLC3	32		
IGG0CLC4	34		
IGG0CLC5	36		
IGG0CLC6	38		
IGG0CLC7	40		
IGG0CLF2	44		
overall organization	21		
path through	62		
selection charts	62		
MODMAP1 data field	57		
MOVEVLV routine in IGG0CLC3	32		
multiple blocks in one index	9		
must-complete function	36		
<b>N</b>			
NAMDELIM data field	58		
NAMDELMP data field	56		
name			
absolute (GDG)	11		
alias	11		
complemented absolute	11		
data set	9		
length	9		
qualified	9		
sequence	9		
simple	9		
NAME			
data field	55		
function	13		
name and content in data areas	47		
NAMLEN data field	56		
NAMLF data field	58		
NAMLG data field	58		
NAMLSTP data Field	56		
NAMTABLE data field	58		
new CVOL pointer entry	48		
new entry, constructing	17		
new extent of SYSCTLG	42		
NEXTCNT data field	56		
NEXTKEY data field	56		
NEXTLVL routine in IGG0CLC2	30		
non-locate			
functions	13		
method of operation	14		
NOP data field	59		
number			
generation	11		
limit of aliases	11		
relative	11		
version	11		
<b>O</b>			
offset in data areas	47		
old CVOL pointer entry	48		
OPEN macro not used	26		

opening a control volume	
method of operation . . . . .	14
module . . . . .	42
operating system . . . . .	9
Operation, Method of . . . . .	14
Operation I, Read . . . . .	14
Operation II, Setup . . . . .	17
Operation III, Write . . . . .	17
options of generation data groups . . . . .	34
OPTNCCW data field . . . . .	59
output from catalog management . . . . .	14
OUTPUT data field . . . . .	59
Overall Organization . . . . .	21
overflow entry . . . . .	17
overview, method of operation . . . . .	14
<b>P</b>	
parameter list . . . . .	13
partitioned data set (PDS) directory	
catalog similar to . . . . .	9
formatting . . . . .	44
path through catalog management modules . . . . .	62
example of . . . . .	23
PDS ( <i>see</i> partitioned data set)	
phases of catalog management . . . . .	21
physical blocks of SYSCTLG . . . . .	9
PKDNUMBR data field . . . . .	58
Program Organization . . . . .	21
PUT routine in IGG0CLC6 . . . . .	38
<b>Q</b>	
qualified name	
example . . . . .	10
introduction . . . . .	9
<b>R</b>	
RC data field . . . . .	59
RD data field . . . . .	59
read operation diagram . . . . .	16
reading blocks of SYSCTLG . . . . .	23
Reading Dumps . . . . .	60
READTTR data field . . . . .	57
RECAT function . . . . .	13
record, data . . . . .	9
REDSGBT data field . . . . .	58
reentrant attribute . . . . .	23
Register Usage . . . . .	24
relating functions to modules . . . . .	62
relative	
converting to absolute track addresses . . . . .	23
generation data group processing . . . . .	28
number (GDG) . . . . .	11
track address (TTR), definition . . . . .	64
RELNUMBER data field . . . . .	58
requirements . . . . .	9
RESALIAS data field . . . . .	58
resource	
enqueueing . . . . .	24
high-level name . . . . .	24
major-minor . . . . .	24
volume index . . . . .	24
volume index control entry . . . . .	24
restriction	
catalog residence . . . . .	10
character dependency . . . . .	24
length of name . . . . .	9
number of aliases . . . . .	11
RETCVOL data field . . . . .	58
RETDATA data field . . . . .	58

ripple	
method of operation . . . . .	17
module . . . . .	38
RKD data filed . . . . .	59
routine	
BLDENTRY in IGG0CLC4 . . . . .	35
BLVCBRTN in IGG0CLC5 . . . . .	36
CALLBLDL	
in IGG0CLC1 . . . . .	29
in IGG0CLC2 . . . . .	30
relation to BLDL . . . . .	23
EMPTYRTN in IGG0CLC5 . . . . .	36
FRBLKRTN in IGG0CLC5 . . . . .	36
FRNDXRTN in IGG0CLC5 . . . . .	36
FRVCBRTN in IGG0CLC5 . . . . .	36
GET	
in IGG0CLC4 . . . . .	35
in IGG0CLC6 . . . . .	38
GETMAINB in IGG0CLC7 . . . . .	40
INCR	
in IGG0CLC4 . . . . .	35
in IGG0CLC5 . . . . .	36
IO	
in IGG0CLC3 . . . . .	33
in IGG0CLC4 . . . . .	35
in IGG0CLC5 . . . . .	37
in IGG0CLC6 . . . . .	38
in IGG0CLC7 . . . . .	41
relation to EXCP . . . . .	24
KEYICE in IGG0CLC3 . . . . .	33
MOVEVL in IGG0CLC3 . . . . .	32
NEXTLVL in IGG0CLC2 . . . . .	30
PUT in IGG0CLC6 . . . . .	38
SCRATCH in IGG0CLC4 . . . . .	35
SETUP	
in IGG0CLC4 . . . . .	35
in IGG0CLC5 . . . . .	36
TOABSL	
in IGG0CLC3 . . . . .	33
in IGG0CLC4 . . . . .	35
in IGG0CLC5 . . . . .	36
in IGG0CLC6 . . . . .	38
relation to IECPCNVT . . . . .	23
TORLTV	
in IGG0CLC2 . . . . .	30
in IGC0CLC3 . . . . .	33
in IGG0CLC5 . . . . .	37
in IGG0CLC6 . . . . .	38
relation to IECPRLTV . . . . .	23
UCATDX in IGG0CLC2 . . . . .	30
WRBLKRTN in IGG0CLC5 . . . . .	36
WRLSTRTN in IGG0CLC5 . . . . .	36
WRTSCRH in IGG0CLC3 . . . . .	32
RESAVE data field . . . . .	54
RPSCCW data field . . . . .	54
RESCNVT data field . . . . .	54
RPSD DSECT data area . . . . .	54
RPSDDKR data field . . . . .	54
RPSINPUT data field . . . . .	54
RPSPTR data field . . . . .	54
RPSR1 data field . . . . .	54
RPSSS data field . . . . .	54
RPSTIC data field . . . . .	54
<b>S</b>	
SAVEAREA data field . . . . .	56
SAVETTR data field . . . . .	57
SAVETTR2 data field . . . . .	57

SCRATCH			
macro instruction	24	UCATDX routine in IGG0CLC2	30
routine in IGG0CLC4	35	UNCAT function	13
SCRPARM data field	58	uncatalog definitions	65
SCRVOLS data field	58	unit control block (UCB)	26
sequence, names	9	updating index control entry	
service modules	21	method of operation	17
services used by catalog management	23	module	40
setup operation	17	using the catalog management routines	12
SETUP routine		<b>V</b>	
in IGG0CLC4	35	VCBVOLCT data field	51
in IGG0CLC5	36	VCBVOLS data field	51
SIDE1 data field	58	VCLSTBLK data field	52
simple name		VFHOLE data field	52
example	10	VHIREC data field	52
introduction	9	VICE	
SKE data field	59	data field	56
supervisor call (SVC)		data area ( <i>see</i> volume index control entry)	
EXIT, used by catalog management	24	VICESAVE data field	56
type for catalog management	9	VLSTBLK data field	52
26	21	VNAME data field	52
28	21	VOLCNT data field	55
supervisor request block (SVRB)	15	VOLSN data field	58
SVRBEXTP data field	57	volume control block (VCB)	
SYSCTLG		definition	65
( <i>see also</i> catalog)		format	51
dumping	60	freeing and writing	36
entry formats	47	introduction	12
example of dumping	61	volume control block pointer entry (VCBPE)	
finding	9	definition	65
formatting	44	format	51
residence	10	introduction	12
system requirements	9	volume index	
system residence, catalog	10	definition	65
<b>T</b>		example	10
task control block (TCB)	15	introduction	9
THETA data field	58	resource enqueueing	24
TIC1 data field	58	volume index control entry (VICE)	
TIC2 data field	59	definition	65
Time Sharing Option (TSO)	12	format	52
TOABSL routine		resource enqueueing	24
in IGG0CLC3	33	updating	
in IGG0CLC4	35	method of operation	17
in IGG0CLC5	37	module	40
in IGG0CLC6	38	volume table of contents (VTOC)	
relation to IECPCNVT	23	definition	65
TORLTV routine		use	15
in IGG0CLC2	30	VTYPE data field	52
in IGG0CLC3	33	<b>W</b>	
in IGG0CLC5	37	WAIT macro instruction	24
in IGG0CLC6	38	WKD data field	59
relation to IECPRLTV	23	work areas, initializing	14
tracing through catalog management modules	64	WORKAREA DSECT data area	
track address conversion	23	format	54
transient area		initializing	26
definition	65	WRBLKRTN routine in IGG0CLC5	36
requirements	9	write operation	17
TRTABLE data field	58	WRITETTR data field	57
TRUE data field	55	writing environment record	40
true name, definition	65	WRLSTRTN routine in IGG0CLC5	36
TSO ( <i>see</i> Time Sharing Option)		WRTSRCH routine in IGG0CLC3	32
TTR data field	55	<b>X</b>	
TYPE data field	55	XCTL macro instruction	29
type 4 SVC	9	XCTLPARM data field	57
UCATDX function	13	2321 Data Cell, restriction	10









**International Business Machines Corporation**  
**Data Processing Division**  
**1133 Westchester Avenue, White Plains, New York 10604**  
**[U.S.A. only]**

**IBM World Trade Corporation**  
**821 United Nations Plaza, New York, New York 10017**  
**[International]**